DOCUMENT RESUME

ED 255 547 TM 850 152

AUTHOR Thissen, David; And Others
TITLE A Computer Program for Simulation Evaluation of IRT Ability Estimators.
INSTITUTION Educational Testing Service, Princeton, NJ. Program Statistics Research Project.
SPONS AGENCY Air Force Human Resources Lab., Brooks AFB, Texas.
REPORT NO ETS-RR-84-37; PSR-TR-84-50
PUB DATE Nov 84
CONTRACT F4168-82-C-0020
NOTE 91p.
PUB TYPE Computer Programs (101)

EDRS PRICE MF01/PC04 Plus Postage.
DESCRIPTORS *Academic Ability; *Computer Simulation; Computer Software; *Estimation (Mathematics); Goodness of Fit; *Item Analysis; *Latent Trait Theory; Mathematical Models
IDENTIFIERS FORTRAN Programing Language; Robustness

ABSTRACT

This report documents a computer program for simulation evaluation of item response theory (IRT) ability estimators. This Honeywell DPS version is a very slight modification of the program originally developed in FORTRAN-77 on a DEC VAX 11/780. The model of the world simulated by the program involves a unidimensional test in which the probability of a correct response is given by a trace line on a single dimension (ability). The appendices contain: (1) the program; (2) a sample of its main output file using four items and 100 simulated examinees at each of five ability levels; and (3) a program to graph the robust simulation results. (Author/BS)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

U.S. DEPARTMENT OF EDUCATION
NATIONAL INSTITUTE OF EDUCATION
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

☐ This document has been reproduced as received from the person or organization originating it.

☐ Minor changes have been made to improve reproduction quality.

• Points of view or opinions stated in this document do not necessarily represent official NIE position or policy.

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY

H. Weidenmiller

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

A Computer Program for Simulation Evaluation of IRT Ability Estimators

David Thissen

Howard Wainer

and

David Rubin

Educational Testing Service
Princeton, New Jersey 08541

Program Statistics Research
Technical Report No. 84-50

Research Report No. 84-37

November 1984

This research was accomplished ancillary to work done under contract F4168-82-C-0020 from the Air Force Human Resources Laboratory of the Educational Testing Service. Howard Wainer, Principal Investigator.

Copyright © 1984 by Educational Testing Service. All Rights Reserved.

The Program Statistics Research Technical Report Series is designed to make the working papers of the Research Statistics Group at Educational Testing Service generally available. The series consists of reports by the members of the Research Statistics Group as well as their external and visiting statistical consultants.

Reproduction of any portion of a Program Statistics Research Technical Report requires the written consent of the author(s).

Table of Contents

		Page
	Introduction	1
I.	A Model of the World	1
II.	The Item Response Data	4
III.	Computation of the Estimates	4
IV.	Results and Output	4
V.	An Example	5
	References	8
	Appendix A	A1-A38
	Appendix B	B1-B33
	Appendix C	C1-C5

Abstract

This report documents the computer program which implements the design of our evaluation by simulation of IRT ability estimators (Wainer and Thissen, 1984). The program is reproduced in Appendix A and a sample of its main output file is Appendix B. It is printed here to facilitate the use of any or all of its routines in future investigations.

A Computer Program for Simulation Evaluation of IRT Ability Estimators

This report documents the computer program which implements the design of our evaluation by simulation of IRT ability estimators (Wainer and Thissen, 1984). The program is reproduced in Appendix A, and a sample of its main output file is Appendix B. The program is printed here to facilitate use of any or all of its routines in future investigations.

The program was originally developed in FORTRAN-77 on a DEC VAX 11/780; very slight modifications produced the Honeywell DPS version included here, and it is anticipated that some or all of the routines may be readily used on any machine with a FORTRAN-77 compiler.

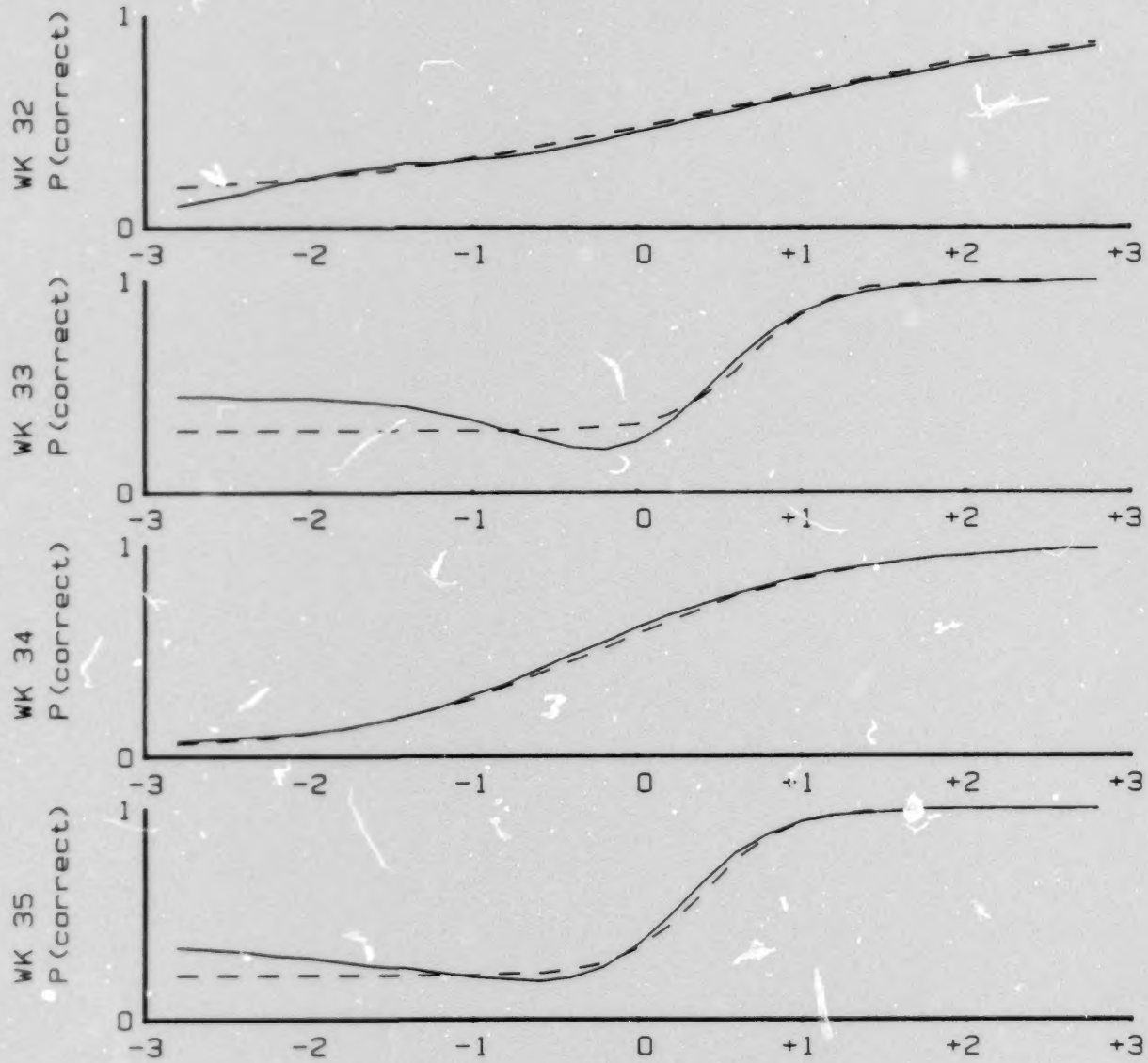
I. A Model of the World

The model of the world simulated by the program involves a test which is unidimensional, that is, the probability of a correct response is given by a trace line on a single dimension (ability). This "true trace line" is of a more complex form than can be represented by any of the binary logistic test models. The trace lines actually used are those given for the correct response by some sets of parameters for the multiple-category IRT model described by Thissen and Steinberg (1984). For four-alternative multiple-choice items, this is a 14-parameter model which includes in its model space a wide variety of possible trace lines. Display 1 shows the four "true" trace lines for the simulation inspired by the performance of four items on the ASVAB Form 8A Word

Knowledge subtest; the parameters are from Thissen and Steinberg (1984). Note that the curves may have relatively exotic shapes, as IRT trace lines go.

In the model of the world simulated here, it is assumed that the item analyst is restricted to the use of conventional logistic IRT models which cannot fit the "true" trace lines exactly. Thus, the IRT model used in ability estimation is always "wrong." The parameters of the n-PL models are estimated from the true trace line as though the true trace line was known, however; so there is no "random error of estimate" in the n-PL item parameters used in the remainder of the simulation. (Jones, Wainer, and Kaplan, 1983, treat the problem of random error in the item parameters using a different approach to the problem.) There is a kind of "bias" inherent in using the wrong model. The bias takes the form of some mis-match between the n-PL fit and the true trace line. In Display 1, the 3-PL trace lines are shown (dashed) along with the true trace line.

In the program, the n-PL trace lines are obtained by fitting the true trace line (represented as a series of 31 probabilities equally spaced on the ability dimension) with the n-PL model by Maximum Likelihood, assuming an infinite sample size and N(0,1) density of examinees. Subroutine READIN obtains the parameters for the multiple category model from file 11 (prepared in advance). Subroutine GENERATE produces the vector of 31 probabilities describing the true trace line, and subroutine NONLINR controls the estimation of the n-PL parameters for subsequent use in ability estimation with the various "wrong models."



Display 1

"True trace lines" (solid) and 3-PL estimates for the four-item WK-inspired example. The fitted 3-PL curves are dashed.

II. The Item Response Data

Five random samples of NSIMU item response vectors are generated by subroutine RESP2. These vectors are generated at five levels of ability (-2, -1, 0, +1, +2) under the usual assumption of conditional independence given the probability of a correct response at the ability level in question; that probability is obtained from the true trace line. To conserve memory, these response vectors are re-generated each time they are needed; otherwise, with long tests, memory requirements become problematic on some (non-virtual) machines.

III. Computation of the Estimates

The value of each estimator (and its estimated standard error) are computed for each response vector for each model in turn: 1-PL (lines in the 100s in DRIVER), 2-PL (in the 200s) and 3-PL (in the 300s). A set of essentially-identically formatted subroutines compute the estimators. These are on pages A-26 to A-35.

IV. Results and Output

Subroutine BINn (n=1,2,3) produces the main body of the output. BIN1 initializes the system. BIN2 stores, for each estimator at each ability level, the number of estimates a specified distance from the true ability level, in "slices" of width 0.2. This is used to produce a display of the distribution of the estimates around the true value; estimates equal to the true value have a distance of zero, of course. The mean difference between the true value and the estimates (bias) and

the standard deviation of the estimates is also computed. The estimated standard errors are similarly placed in bins for semi-graphic presentation. The results are placed in output file 12.

Subroutine WRIT15 creates an output file (on 15) containing the bias, standard deviations, and RMSE for each estimator at each ability level under each of the scaling conventions described by Wainer and Thissen (1984). The data in file 15 are the input for the graphical display programs such as GRAPHOUT (Appendix C) and the XTREE series (Thissen and Wainer, 1984).

V. An Example

Appendix B reproduces the output of the program for a sample run. The example uses four items (with the true trace lines of Display 1, above) and 100 simulees at each level of ability. Page B-1 gives the title and control information. The parameters for the multiple category model are given on page B-2, along with the true probabilities at each of the five levels of ability (5 Ps). The parameters of the 1-PL model most closely approximating the true trace lines are given on page B-3.

Page B-4 gives the results for the mean (or EAP) estimator with the 1-PL model. (A text editor has been used to remove counts of zero from the output, leaving a semi-graphic.) There are 100 response vectors represented in each column; therefore the numbers are percentages. The numbers at the upper right mean that, for true ability +2, one of the estimators was more than 1.8 away from the true value, 26 were between 1.4 and 1.6 away (these estimates averaged about 0.5) and 73 of the estimates were between 0.8 and 1.0 away (these averaged about 1.1). The

average difference between the estimates and 2.0 was 1.06; so the average of the estimates was about 0.94. With four items, "regression to the mean" dominates the results for extreme abilities. A four-item set can provide only limited evidence that an examinee is of very high or very low ability; the 1.1 estimate is the EAP result for vectors in which all four items are correct -- no response pattern scores higher. On page B-4, the lower panel gives the distributions of the estimated standard errors; all are between 0.7 and 0.8.

With four items, the displays in Appendix B are dominated by regression to the mean. Display 2 shows an example of BIN3 ouput for a 40-item simulation in which the estimator gives much less biased, lower-standard error performance. The outcome in Display 2 is about the best that can be obtained with IRT: the top panel is (nearly) a straight line around zero bias, and the lower panel shows small standard erros (about 0.3), which match the actual standard deiviations of the estimators. Page B-4 displays fairly poor performance.

Pages B-5 - B-13 ar similar displays for the remaining 1-PL estimators; pp. B-15 - B-23 are for the 2-PL estimators and pp. B-25 - B-33 are for the 3-PL estimators.

Robust Estimation of Ability

Simulation Program - 7

4 LAST GS X 10 Estimator= MEAN

Binned u-uhat

Bin End	u=-2.0	u=-1.0	u= 0.0	u= 1.0	u= 2.0
1.6- 1.8					
1.4- 1.6					
1.2- 1.4					
1.0- 1.2		1			
0.8- 1.0		1			1
0.6- 0.8		4			2
0.4- 0.6		10	1	7	18
0.2- 0.4		18	17	19	35
0.0- 0.2	10	20	41	27	18
-0.2- 0.0	26	21	30	33	11
-0.4--0.2	20	19	10	12	15
-0.6--0.4	23	1		2	
-0.8--0.6	12	5	1		
-1.0--0.8	8				
-1.2--1.0	1				
-1.4--1.2					
-1.6--1.4					
-1.8--1.6					
End					
Mean d	-0.358	0.055	0.033	0.034	-0.185
Std. Dev.	0.283	0.345	0.195	0.218	0.290

Binned Estimated s.e.

Bin End	u=-2.0	u=-1.0	u= 0.0	u= 1.0	u= 2.0
1.9- 1.8					
1.8- 1.7					
1.7- 1.6					
1.6- 1.5					
1.5- 1.4					
1.4- 1.3					
1.3- 1.2					
1.2- 1.1					
1.1- 1.0					
1.0- 0.9					
0.9- 0.8					
0.8- 0.7					
0.7- 0.6					
0.6- 0.5	6				
0.5- 0.4	75	14			15
0.4- 0.3	17	44			39
0.3- 0.2	2	42	87	90	46
0.2- 0.1			13	10	
0.1- 0.0					
Mean s.e.	0.445	0.318	0.217	0.222	0.327
Std. Dev.	0.057	0.070	0.016	0.016	0.070

Display 2

References

Jones, D.H., Wainer, H. & Kaplan, B. Estimating ability with three item response models when the models are wrong and their parameters are inaccurate. Ms. in preparation, 1983.

Thissen, D. & Steinberg, L. A model for multiple choice items. Psychometrika, in press, 1984.

Thissen, D. & Wainer, H. The graphical display of simulation results, with applications to the comparison of robust IRT estimators of ability. Ms. in preparation, 1984.

Wainer, H. & Thissen, D. Estimating ability with the wrong model. Ms. in preparation, 1984.

Appendix A

```
      PROGRAM DRIVER
C  PROGRAM FOR SIMULATION EVALUATION OF ABILITY ESTIMATORS
C  BY
C  DAVID THISSEN AND DAVID RUBIN
C
C  THIS IS THE HONEYWELL VERSION, WHICH RUNS ON HONEYWELL DPS
C  SYSTEMS. IT IS WRITTEN IN FORTRAN-77.
C  THIS (HONEYWELL) VERSION USES IMSL
C      SUBROUTINE GGUBFS TO GENERATE
C  RANDOM NUMBERS.
C
C  THE PROGRAM USES A NUMBER OF I/O FILES-
C    05 IS A CONTROL INPUT FILE. IT MAY BE THE TERMINAL OR A DISK FILE.
C   IT CONTAINS A ONE-LINE TITLE, AND THE RESPONSES (ONE PER LINE)
C   TO THE QUESTIONS ASKED IN
C      SUBROUTINE QUERY AND JUST BELOW THE
C   CALL TO QUERY IN DRIVER.
C    06 IS AN OUTPUT FILE. WHEN THE PROCRAM IS USED IN A TIME-
C   SHARING ENVIRONMENT, IT PUTS PROMPTS ON THE TERMINAL.
C    11 IS AN INPUT FILE, WHICH MUST CONTAIN THE PARAMETERS FOR
C   THE BSBT MODEL. THE PARAMETERS (A'S, C'S, AND D'S IN THE
C   NOTATION OF THE THISSEN AND STEINBERG PAPER) ARE READ IN FREE
C   FORMAT, ONE ITEM AT A TIME  AND WITHIN THAT ONE CATEGORY
C   AT A TIME. CATEGORY 0 IS READ FIRST; IT HAS NO D; THEN CATEGORIES
C   1,2,3 AND 4.
C    12 IS THE MAIN OUTPUT FILE, WHICH CONTAINS TABLES OF THE COUNTS
C   OF BINNED ESTIMATORS AND OTHER STATISTICS. IT MAY BE DIRECTED
C   TO A LINE PRINTER OR EDITTED TO PRODUCE MORE FORMAL OUTPUT.
C   IF IT IS EDITTED, IT IS WISE TO REPLACE ALL THE ZEROES " 0 "
C   WITH BLANKS. IT CLEANS UP THE OUTPUT A LOT.
C    13 IS A SMALL OUTPUT FILE WHICH CONTAINS THE TRUE
C   TRACE LINE (IN THE FORM OF A SERIES OF P'S ACROSS U) AND THE FITTED
C   1-PL, 2-PL, AND 3-PL TRACE LINES. THIS COULD BE USED FOR GRAPHICS.
C    14 IS A SCRATCH FILE.
C    15 IS AN OUTPUT FILE CONTAINING THE MEAN DEVIATIONS AND STANDARD
C   DEVIATIONS OF EACH ESTIMATOR AT EACH LEVEL OF U. IT IS USED AS INPUT
C   FOR THE ASSOCIATED GRAPHICS PROGRAMS-SPECIFICALLY THE TREE#
C   SERIES AND GRAPHOUT. IT IS FORMATTED TO BE MACHINE-, NOT HUMAN-
C   READABLE.
C
C  USAGE OF THE PROGRAM:
C
C  1. FOR SOME NUMBER OF ITEMS (UP TO 40) PARAMETERS OF THE BSBT
C     MODEL WHICH PRODUCE THE DESIRED TRACE LINES FOR CORRECT RESPONSE
C     ARE DETERMINED AND PLACED IN THE FILE TO BE CALLED "11".
C  2. THE PROGRAM IS RUN, WITH ANSWERS TO A SERIES OF QUESTIONS
C     MADE AVAILABLE ON FILE 05-
C      WHAT IS THE NAME OF THE TEST?
```

```
C       NUMBER OF QUADRATURE POINTS FOR MEANS (LIMIT IS 100)?
C       CUT-OFF POINT FOR BIWEIGHTS?
C       PRIOR DISTRIBUTION? (RECTANGLE, OR 0=N(0,1)
C       FOR LOTS OF PRINTOUT, TYPE 1? (THIS INCREASES OUTPUT ON 12)
C       HERE THE USER HAS THE OPPORTUNITY TO PUT THE BSBT PARAMETERS
C        IN, IF THEY ARE NOT ALREADY IN FILE 11.
C       ENTER THE NUMBER OF SIMULEES?
C       ENTER THE SEED FOR THE RANDOM NUMBER GENERATOR?
C
C  3. THE PROGRAM PRODUCES THE OUTPUT DESCRIBED IN THE DOCUMENTATION.
C
C
C
      REAL*8 SEED1,SEED
        INTEGER FLAG,RESP(40,1,1)
        LOGICAL ERROR
        DIMENSION DER1(31,40),UG(31),PG(31,40)
        DIMENSION DER2(31,40),DER3(31,40),UHAT(10),SE(1 )
        DIMENSION A1(40),D1(40),C1(40),PROB(5,40)
        DIMENSION U(5),WT(31),PSEUDO(40),W(4)
        REAL MLE(31,40),JACOB(600),INVER(600),INV1(41, 1)
        DIMENSION AK(40,5),CK(40,5),DK(40,5)
        CHARACTER*16 NAME
        DATA U / -2 ,-1.0, 0.0, 1.0, 2.0 /
C
C       CALL FXOPT(19,1,1,0)
        NCOL=40
        NROW=1
        NSUBJ=31
        NBIG=1
        NLVL=5
        UG(1)=-3.0
        DO 10 I=2,NSUBJ
10      UG(I)=UG(I-1)+6./(NSUBJ-1)
        CALL QUERY(NAME,NITEM,W,LOTS)
        CALL READIN(NITEM,NCOL,AK,CK,DK,ERROR)
        WRITE(6,20)
 20     FORMAT(' ENTER THE NUMBER OF SIMULEES:')
        READ(5,*) NSIMU
      WRITE(6,21)
 21   FORMAT(' ENTER SEED')
      READ(5,*) SEED1
        WRITE(12,11) NAME,NITEM,W(1),W(3),W(4),LOTS,NSIMU,SEED1
11      FORMAT('1ROBUST ABILITY ESTIMATION PROJECT--',A16/
     &  '0A TEST OF ',I3,' ITEMS.'/
     &  '0MEANS USE ',F3.0,' QUADRATURE POINTS.'/
     &  '0CUT-OFF FOR BIWEIGHT IS ',F4.2/
     &  '0PRIOR CODE IS (VALUE=RECTANGLE BOUNDARY,0=NORMAL) ',F3.1/
     &  '0LOTS=',I1/
     &  '0NUMBER OF SIMULATED EXAMINEES=',I5/'0SEED=',F12.0)
        CALL GENERATE(NSUBJ,NITEM,NSUBJ,NCOL,AK,CK,DK,UG,PG)
        WRITE(13,13) NAME
13      FORMAT('1ACTUAL ICCS FOR--'A16/' ')
```

```
      DO 17 J=1,NITEM
      WRITE(13,14) J,(PG(K,J),K=1,31)
14    FORMAT(' ITEM',I3,15F5.2/3X,16F5.2)
      I1=6
      DO 16 I=1,5
      PROB(I,J)=PG(I1,J)
      I1=I1+5
16    CONTINUE
17    CONTINUE
      WRITE(12,18) NAME
18    FORMAT('1SOME OF THE BASIC STUFF FOR THIS RUN-'A16)
      DO 195 J=1,NITEM
      WRITE(12,19) J,(AK(J,K),K=1,5),(CK(J,K),K=1,5),(DK(J,K),K=2,5),
     & ( PROB(I,J),I=1,5)
19    FORMAT('0ITEM',I3/' AK ',5F7.3/' CK ',5F7.3/' DK ',7X,4F7.3/
     & '05 PS ',F5.3,4F7.3)
195   CONTINUE
      IF(LOTS.EQ.0) GOTO 29
      WRITE(12,22) NAME
22    FORMAT('1',A16/'0GENERATED RESPONSE VECTORS-')
      SEED=SEED1
      DO 25 J=1,NSIMU
      CALL RESP2(NROW,NCOL,NITEM,NLVL,1,1,U,PROB,RESP,SEED,I)
      WRITE(12,24) U(I),J,(RESP(K,1,1),K=1,NITEM)
24    FORMAT(1X,F4.1,1X,I4,1X,100I1)
25    CONTINUE
29      DO 40 J=1,NITEM
          D1(J)=0.0
          A1(J)=1.0
          C1(J)=0.0
40      CONTINUE
        FLAG=1
        CALL NONLINR(FLAG,LOG,PG,NITEM,NSUBJ,NSUBJ,DER1,
     $           DER2,DER3,WT,JACOB,PSEUDO,INVER,MLE,C1,A1,D1,INV1)
        WRITE(13,131) FLAG,NAME
131     FORMAT('1',I1,'-PL ICCS FOR--',A16/' ')
        DO 105 J=2,NITEM
105     A1(J)=A1(1)
        WRITE(12,110) NAME
110     FORMAT('1ESTIMATED 1-PL PARAMETERS--',A16/
     &  '0ITEM   LOCATION    SLOPE'/' ')
        DO 112 J=1,NITEM
        WRITE(13,14) J,(MLE(K,J),K=1,31)
112     WRITE(12,113) J,D1(J),A1(J)
113     FORMAT(1X,I3,1X,3F10.3)
        IF(LOTS.NE.0) WRITE(12,114) FLAG,NAME
114   FORMAT('1',I1,'-PL U-ESTIMATES ',A16/'0'13X,'MEAN',4X,'HMEAN.25',
     &  2X,'HMEAN.5',3X,'HMEAN1.0',4X,'MODE',5X,'HEST0.5',3X,'HEST1.0',
     &  3X,'HEST2.0',5X,'BIWT',4X,'AMJACK')
        CALL AMINIT(NITEM,RESP,D1,A1,C1,UHAT,SE,W)
        CALL BIN1(10)
      SEED=SEED1
```

```
        DO 150 I=1,5
        DO 140 J=1,NSIMU
      CALL RESP2(NROW,NCOL,NITEM,NLVL,NBIG,1,U,PROB,RESP,SEED,I)
        CALL MEAN (NITEM,RESP(1,1,1),D1,A1,C1,UHAT(1),SE(1),W)
        DO 120 K=2,10
120     UHAT(K)=UHAT(1)
        W(2)=0.25
        CALL HMEAN (NITEM,RESP(1,1,1),D1,A1,C1,UHAT(2),SE(2),W)
        W(2)=0.5
        CALL HMEAN (NITEM,RESP(1,1,1),D1,A1,C1,UHAT(3),SE(3),W)
        W(2)=1.0
        CALL HMEAN (NITEM,RESP(1,1,1),D1,A1,C1,UHAT(4),SE(4),W)
        CALL MODE (NITEM,RESP(1,1,1),D1,A1,C1,UHAT(5),SE(5),W)
        W(2)=0.5
        CALL HEST (NITEM,RESP(1,1,1),D1,A1,C1,UHAT(6),SE(6),W)
        W(2)=1.0
        CALL HEST (NITEM,RESP(1,1,1),D1,A1,C1,UHAT(7),SE(7),W)
        W(2)=2.0
        CALL HEST (NITEM,RESP(1,1,1),D1,A1,C1,UHAT(8),SE(8),W)
        CALL BIWT (NITEM,RESP(1,1,1),D1,A1,C1,UHAT(9),SE(9),W)
        CALL AMJACK (NITEM,RESP(1,1,1),D1,A1,C1,UHAT(10),SE(10),W)
        CALL BIN2(U(I),UHAT,SE,10)
      IF(LOTS.EQ.0) GOTO 140
      WRITE(12,135) U(I),J,(UHAT(K),SE(K),K=1,10)
135   FORMAT(1X,F4.1,1X,I4,10(F5.1,'(',F3.1,')'))
140     CONTINUE
150     CONTINUE
        CALL BIN3(NAME,10,NSIMU)
C
        FLAG=2
        DO 201 J=1,NITEM
201     CALL NONLINR(FLAG,UG,PG(1,J),1,NSUBJ,NSUBJ,DER1,
     $ DER2,DER3,WT,JACOB,PSEUDO,INVER,MLE(1,J),C1(J),A1(J),D1(J),INV1)
        WRITE(13,121) FLAG,NAME
        WRITE(12,210) NAME
210     FORMAT('1ESTIMATED 2-PL PARAMETERS--',A16/
     &  '0ITEM   LOCATION    SLOPE'/' ')
        DO 212 J=1,NITEM
        WRITE(13,14) J,(MLE(K,J),K=1,31)
212     WRITE(12,113) J,D1(J),A1(J)
        IF(LOTS.NE.0) WRITE(12,214) FLAG,NAME
214   FORMAT('1',I1,'-PL U-ESTIMATES-',A16/'0'13X,'MEAN',4X,'HMEAN.25',
     &  2X,'HMEAN.5',3X,'HMEAN1.0',4X,'MODE',5X,'HEST0.5',3X,'HEST1.0',
     &  3X,'HEST2.0',5X,'BIWT')
        CALL BIN1(9)
      SEED=SEED1
        DO 250 I=1,5
        DO 240 J=1,NSIMU
      CALL RESP2(NROW,NCOL,NITEM,NLVL,NBIG,1,U,PROB,RESP,SEED,I)
        CALL MEAN (NITEM,RESP(1,1,1),D1,A1,C1,UHAT(1),SE(1),W)
        DO 220 K=2,9
220     UHAT(K)=UHAT(1)
        W(2)=0.25
```

```
      CALL HMEAN (NITEM,RESP(1,1,1),D1,A1,C1,UHAT(2),SE(2),W)
      W(2)=0.5
      CALL HMEAN (NITEM,RESP(1,1,1),D1,A1,C1,UHAT(3),SE(3),W)
      W(2)=1.0
      CALL HMEAN (NITEM,RESP(1,1,1),D1,A1,C1,UHAT(4),SE(4),W)
      CALL MODE (NITEM,RESP(1,1,1),D1,A1,C1,UHAT(5),SE(5),W)
      W(2)=0.5
      CALL HEST (NITEM,RESP(1,1,1),D1,A1,C1,UHAT(6),SE(6),W)
      W(2)=1.0
      CALL HEST (NITEM,RESP(1,1,1),D1,A1,C1,UHAT(7),SE(7),W)
      W(2)=2.0
      CALL HEST (NITEM,RESP(1,1,1),D1,A1,C1,UHAT(8),SE(8),W)
      CALL BIWT (NITEM,RESP(1,1,1),D1,A1,C1,UHAT(9),SE(9),W)
      CALL BIN2(U(I),UHAT,SE,9)
     IF(LOTS.EQ.0) GOTO 240
     WRITE(12,135) U(I),J,(UHAT(K),SE(K),K=1,9)
240   CONTINUE
250   CONTINUE
      CALL BIN3(NAME,9,NSIMU)
C
      FLAG=3
      DO 301 J=1,NITEM
301   CALL NONLINR(FLAG,UG,PG(1,J),1,NSUBJ,NSUBJ,DER1,
     & DER2,DER3,WT,JACOB,PSEUDO,INVER,MLE(1,J),C1(J),A1(J),D1(J),INV1)
      WRITE(13,131) FLAG,NAME
      WRITE(12,310) NAME
310   FORMAT('1ESTIMATED 3-PL PARAMETERS--',A16/
     & '0ITEM   LOCATION    SLOPE  LWR ASYMP'/' ')
      DO 312 J=1,NITEM
      WRITE(13,14) J,(MLE(K,J),K=1,31)
312   WRITE(12,113) J,D1(J),A1(J),C1(J)
      IF(LOTS.NE.0) WRITE(12,214) FLAG,NAME
      CALL BIN1(9)
     SEED=SEED1
      DO 350 I=1,5
      DO 340 J=1,NSIMU
     CALL RESP2(NROW,NCOL,NITEM,NLVL,NBIG,1,U,PROB,RESP,SEED,I)
      CALL MEAN (NITEM,RESP(1,1,1),D1,A1,C1,UHAT(1),SE(1),W)
      DO 320 K=2,9
320   UHAT(K)=UHAT(1)
      W(2)=0.25
      CALL HMEAN (NITEM,RESP(1,1,1),D1,A1,C1,UHAT(2),SE(2),W)
      W(2)=0.5
      CALL HMEAN (NITEM,RESP(1,1,1),D1,A1,C1,UHAT(3),SE(3),W)
      W(2)=1.0
      CALL HMEAN (NITEM,RESP(1,1,1),D1,A1,C1,UHAT(4),SE(4),W)
      CALL MODE (NITEM,RESP(1,1,1),D1,A1,C1,UHAT(5),SE(5),W)
      W(2)=0.5
      CALL HEST (NITEM,RESP(1,1,1),D1,A1,C1,UHAT(6),SE(6),W)
      W(2)=1.0
      CALL HEST (NITEM,RESP(1,1,1),D1,A1,C1,UHAT(7),SE(7),W)
      W(2)=2.0
      CALL HEST (NITEM,RESP(1,1,1),D1,A1,C1,UHAT(8),SE(8),W)
```

```
        CALL BIWT (NITEM,RESP(1,1,1),D1,A1,C1,UHAT(9),SE(9),W)
        CALL BIN2(U(I),UHAT,SE,9)
      IF(LOTS.EQ.0) GOTO 340
      WRITE(12,135) U(I),J,(UHAT(K),SE(K),K=1,9)
340     CONTINUE
350     CONTINUE
        CALL BIN3(NAME,9,NSIMU)
C
        CALL WRIT15
        STOP
        END
```

```
      SUBROUTINE QUERY(NAME,NITEM,W,LOTS)
C
C       QUERY ASKS THE USER TO PROVIDE CERTAIN VALUES.
C
C       NAME IS THE ALPHANUMERIC NAME OF THE TEST.
C       NITEM IS THE NUMBER OF ITEMS IN THE TEST.
C       W IS THE VECTOR OF WORKING PARAMETERS.
C
        CHARACTER*16 NAME
        DIMENSION W(10),AK(5),CK(5),DK(5)
        WRITE (6,2001)
 2001   FORMAT (' NAME OF TEST?')
        READ (5,2002) NAME
 2002   FORMAT (A16)
        WRITE (6,2004)
 2004   FORMAT (' NUMBER OF QUADRATURE POINTS FOR MEANS?')
        READ (5,*) W(1)
        WRITE (6,2005)
 2005   FORMAT (' CUT-OFF POINT FOR BIWEIGHTS?')
        READ (5,*) W(3)
        WRITE (6,2006)
 2006   FORMAT (' PRIOR DISTRIBUTION? RECTANGLE BOUND,OR 0=N(0,1)')
        READ (5,*) W(4)
        WRITE(6,20065)
20065   FORMAT(' FOR LOTS OF PRINTOUT, TYPE 1 (ELSE 0)')
        READ(5,*) LOTS
        WRITE(6,2007)
 2007   FORMAT(' IF THE B-S PARAMETERS ARE ON FOR011 ALREADY, ENTER 0.'/
     & ' OTHERWISE, ENTER THE NUMBER OF ITEMS IN THE TEST.')
        READ(5,*) NITEM
        IF(NITEM.EQ.0) RETURN
        WRITE(11,*) NITEM
        DO 2020 I=1,NITEM
        WRITE(6,2008) I
 2008   FORMAT(' ENTER PARAMETERS FOR ITEM',I3/' 5 AK S?')
        READ(5,*) AK
        WRITE(6,2009)
 2009   FORMAT(' 5 CK S?')
        READ(5,*) CK
        WRITE(6,2010)
 2010   FORMAT(' 4 DK S?')
        DK(1)=0.0
        READ(5,*) (DK(K),K=2,5)
        DO 2015 K=1,5
        WRITE(11,2012) AK(K),CK(K),DK(K)
 2012   FORMAT(3F10.4)
 2015   CONTINUE
 2020   CONTINUE
        RETURN
        END
```

```
      SUBROUTINE READIN(NITEM,NCOL,AK,CK,DK,ERROR)
C
C       READIN READS THE PARAMETERS NECESSARY TO GENERATE
C         PROBABILITIES FROM THE B-S MODEL.
C
C       NCOL IS THE MAXIMUM NUMBER OF COLUMNS (I.E., ITEMS).
C       ERROR IS A LOGICAL FLAG INDICATING WHETHER OR NOT THE
C         PARAMETERS VIOLATE THE ASSUMPTIONS OF THE B-S MODEL.
C       AK, CK, AND DK ARE NCOL X 5 ARRAYS OF THE PARAMETERS OF
C         THE B-S MODEL. NOTE THAT 5 REPRESENTS THE NUMBER OF
C         CHOICES ON EACH TEST ITEM, AND THAT DK(I,1) IS ZERO
C         FOR ALL I (I.E., FOR ALL ITEMS), AND THUS NEEDN'T
C         CONCERN US.
C       SUMAK, SUMCK, AND SUMDK ARE THE SUM OF THEIR RESPECTIVE
C         ARRAYS (AK,CK, OR DK) OVER ALL CHOICES FOR A FIXED ITEM.
C       I INDICATES THE ITEM BEING CONSIDERED.
C       K INDICATES WHICH CHOICE OF THAT ITEM IS BEING CONSIDERED.
C
        LOGICAL ERROR
        DIMENSION AK(NCOL,5),CK(NCOL,5),DK(NCOL,5)
        REWIND 11
        READ(11,*) NITEM
        DO 10 I=1,NITEM
          READ(11,*) AK(I,1),CK(I,1)
          SUMAK=AK(I,1)
          SUMCK=CK(I,1)
          SUMDK=0.0
          DO 11 K=2,5
            READ(11,*) AK(I,K),CK(I,K),DK(I,K)
            SUMAK=SUMAK+AK(I,K)
            SUMCK=SUMCK+CK(I,K)
            SUMDK=SUMDK+DK(I,K)
11        CONTINUE
          ERROR=(ABS(SUMAK) .GE. .01) .OR. (ABS(SUMCK) .GE. .01)
     $            .OR. (ABS(SUMDK-1.0) .GE. .01)
          IF (ERROR) THEN
            WRITE(6,989) I
989   FORMAT(' READ-IN INTERRUPTED. PARAMETERS DID NOT SUM TO ZERO.',I3)
            STOP
          END IF
10      CONTINUE
12      RETURN
        END
```

```
      SUBROUTINE GENERATE(NSUBJ,NITEM,NROW,NCOL,AK,CK,DK,U,PROB)
C
C       GENERATE GENERATES PROBABILITIES FOR THE B-S MODEL.
C
C       DENOM1 AND DENOM2 SUM UP THE COMPONENTS OF TWO SUMS WHICH
C         ARE DIVISORS IN THE B-S EXPRESSION OF PROBABILITY
C         OF ITEM RESPONSE.
C       FOR DESCRIPTIONS OF THE VARIABLES NCOL,AK,CK, AND DK,
C         SEE
C      SUBROUTINE READIN; FOR NSUBJ,NITEM, OR NROW, SEE
C         NONLINR.
C
        DIMENSION AK(NCOL,5),CK(NCOL,5),DK(NCOL,5)
        DIMENSION U(NSUBJ),PROB(NROW,NITEM)
        HK=1.0
        HKSTAR=1.0
        DO 20 I=1,NSUBJ
          DO 21 J=1,NITEM
            DENOM1=EXP(AK(J,1)*U(I)+CK(J,1))
            DO 22 K=2,5
              DENOM1=DENOM1+EXP(AK(J,K)*U(I)+CK(J,K))
22          CONTINUE
            PROB(I,J)=(HKSTAR*EXP(AK(J,5)*U(I)+CK(J,5))
     &                  +HK*DK(J,5)
     $                   *EXP(AK(J,1)*U(I)+CK(J,1)))/DENOM1
21        CONTINUE
20      CONTINUE
        RETURN
        END
```

```
      SUBROUTINE RESP2(NROW,NCOL,NITEM,NLVL,NBIG,NSIMU,U,PROB,RESP,
     & SEED,IT)
C
C       RESP2 GENERATES RESPONSE DATA BASED UPON GIVEN PROBABILITIES.
C
C       NROW IS THE MAXIMUM NUMBER OF ABILITY LEVELS.
C       NITEM IS THE NUMBER OF ITEMS.
C       NLVL IS THE NUMBER OF ABILITY LEVELS.
C       NSIMU IS THE NUMBER OF SIMULEES FOR EACH ABILITY.
C       U IS THE NSUBJ LONG VECTOR OF SUBJECT ABILITIES.
C       PROB IS THE NSUBJ X NITEM ARRAY OF THE PROBABILITY THAT
C         THE ITH "ABILITY" ANSWERS THE JTH ITEM CORRECTLY.
C       RESP IS THE NITEM ARRAY WHERE EACH ENTRY REPRESENTS WHETHER
C         THE PRESENT SUBJECT ANSWERED THE JTH ITEM CORRECTLY
C         (1=CORRECT, 0=INCORRECT).
C       I IS THE PRESENT "ABILITY".
C       J IS THE PRESENT ITEM.
C       K IS THE PRESENT SIMULATION.
C       SEED1 AND SEED2 ARE SEEDS FOR GENERATING RANDOM NUMBERS.
C
      REAL*8 SEED
        INTEGER RESP(NCOL,NBIG,NROW),SEED1,SEED2
        DIMENSION U(NLVL),PROB(NLVL,NCOL)
C         WRITE(6,960)
C 960     FORMAT(' ENTER TWO SEED INTEGERS:')
C         READ(5,*) SEED1,SEED2
C         WRITE(12,961) SEED1,SEED2
C 961     FORMAT('0THE SEEDS WERE ',2I20)
        DO 30 I=1,NROW
          DO 31 K=1,NSIMU
            DO 32 J=1,NITEM
              IF (GGUBFS(SEED) .LT. PROB(IT,J)) THEN
                RESP(J,K,I)=1
              ELSE
                RESP(J,K,I)=0
              END IF
 32         CONTINUE
 31       CONTINUE
 30     CONTINUE
        RETURN
        END
```

```
      SUBROUTINE NONLINR(FLAG,U,PROB,NITEM,NSUBJ,NROW,
     $      DER1,DER2,DER3,WT,JACOB,PSEUDO,INVER,MLE,C,A,D,INV1)
C
C  NONLINR PERFORMS NONLINEAR MAXIMUM LIKELIHOOD REGRESSION
C    TO ESTIMATE THE PARAMETERS OF A LOGISTIC MODEL.
C
C  THE SUBPROGRAMS PERFORM THE FOLLOWING FUNCTIONS:
C    WEIGHT PROVIDES WEIGHTS FOR THE REGRESSION.
C    RAPHSON USES THE NEWTON-RAPHSON METHOD TO ESTIMATE THE
C       PARAMETERS OF INTEREST.
C
C  THE VARIABLES ARE:
C    NSUBJ IS THE NUMBER OF ABILITY LEVELS.
C    NITEM IS THE NUMBER OF ITEMS.
C    NROW IS THE MAXIMUM NUMBER OF SUBJECTS.
C    FLAG IS AN INDICATOR OF THE MODEL ASSUMING VALUES OF 1,2,OR 3;
C       "N" INDICATES AN "N"-PARAMETER LOGISTIC MODEL.
C    PROB IS AN NSUBJ X NITEM MATRIX OF PROBABILITIES:
C      PROB(I,J) IS THE (EMPIRICAL OR GIVEN) PROPORTION OF THE ITH
C      ABILITY LEVEL ANSWERING THE JTH ITEM CORRECTLY.
C    U IS AN NSUBJ COLUMN VECTOR OF "ABILITY".
C    WT IS AN NSUBJ COLUMN VECTOR OF WEIGHTS.
C    C IS AN NITEM VECTOR OF (ESTIMATED/GIVEN) ASYMTOPE PARAMETERS
C    A IS AN NITEM VECTOR OF ESTIMATED SLOPE PARAMETERS FOR TWO OR
C      THREE PARAMETER LOGISTICS. FOR THE ONE PARAMETER LOGISTIC,
C      ONLY A(1) IS ESTIMATED; IT IS THE COMMON SLOPE FOR ALL ITEMS.
C    D IS AN NITEM VECTOR OF ESTIMATED DIFFICULTY PARAMETERS.
C    C, A, AND D ARE ALL ROW VECTORS.
C
C  FOR DISCUSSION OF THE VARIABLES JACOB, PSEUDO, INVER, INV1,
C    DER1, DER2, DER3, AND MLE, SEE THE COMMENTS IN
C      SUBROUTINE RAPHSON.
C
       INTEGER FLAG
       DIMENSION PROB(NROW,NITEM), U(NSUBJ), WT(NSUBJ)
       DIMENSION C(NITEM), A(NITEM), D(NITEM)
       REAL JACOB((FLAG**2+FLAG)*NITEM/2), MLE(NROW,NITEM)
       REAL INVER((FLAG**2+FLAG)*NITEM/2)
       DIMENSION DER1(NROW,NITEM),DER2(NROW,NITEM),DER3(NROW,NITEM)
       CALL WEIGHT (NSUBJ,U,WT)
       CALL RAPHSON (FLAG,NITEM,NSUBJ,NROW,PROB,U,WT,C,A,D,
     $                 DER1,DER2,DER3,JACOB,PSEUDO,INVER,MLE,INV1)
       RETURN
       END
```

```
      SUBROUTINE WEIGHT (NSUBJ,U,WT)
C
C  WEIGHT ASSIGNS WEIGHTS TO ABILITY LEVELS.
C  IT ASSUMES THEY ARE DISTRIBUTED AS N(0,1).
C
        DIMENSION U(NSUBJ), WT(NSUBJ)
        DO 400 KOUNT=1,NSUBJ
          WT(KOUNT) = EXP(-U(KOUNT)**2/2.0)
 400    CONTINUE
        RETURN
        END
```

```
      SUBROUTINE RAPHSON (FLAG,NITEM,NSUBJ,NROW,PROB,U,WT,C,A,D,
     $  DER1,DER2,DER3,JACOB,PSEUDO,INVER,MLE,INV1)
C
C  RAPHSON USES THE NEWTON-RAPHSON METHOD TO FIND THE SOLUTION TO
C    THE CRITICAL EQUATIONS OF THE MAXIMUM LIKELIHOOD REGRESSION
C    FOR THE ONE-,TWO-,OR THREE-PARAMETER LOGISTIC MODEL.
C    THE FIRST SECTION OF THE
      SUBROUTINE IS USED FOR THE ONE-
C    PARAMETER MODEL (AND HAS STATEMENT LABELS FROM 100-199),
C    THE SECOND FOR THE TWO-PARAMETER MODEL (200-299), AND THE
C    THIRD FOR THE THREE (300-399).
C  IN ANY SECTION, THE PROCEDURE IS AS FOLLOWS:
C    1) USING THE INITIAL VALUES OF THE D, A, AND C VECTORS,
C         INITIALIZE THE MLE MATRIX AND ITS ASSOCIATED DERIVATIVES.
C    2) CALCULATE THE JACOBIAN (I.E. HESSIAN).
C    3) CALCULATE ITS INVERSE.
C    4) CALCULATE THE FUNCTION TO BE ZEROED (I.E. THE GRADIENT).
C    5) IF STEP 4 IS TOO LARGE, AND LESS THAN 100 ITERATIONS HAVE BEEN
C         USED SO FAR, REESTIMATE THE PARAMETERS, RECALCULATE
C         THE MLE AND ITS DERIVATIVES, AND RETURN TO STEP 2.
C    6) IF STEP 4 IS SUFFICIENTLY SMALL, EXECUTION ENDS SUCCESSFULLY.
C         IF 100 ITERATIONS HAVE BEEN USED AND STEP 4 IS TOO LARGE,
C         EXECUTION ENDS UNSUCCESSFULLY.
C  THE EQUATION OF WHICH WE ARE ATTEMPTING TO FIND THE ZERO IS
C
C            (F1(D))
C    F(X) = (F2(A)) = GRADIENT(LOG LIKELIHOOD)
C            (F3(C))
C
C    THUS THE JACOBIAN OF THE GRADIENT IS THE HESSIAN OF THE LIKELIHOOD.
C  THE  ONE AND TWO PARAMETER MODELS NEGLECT C AND F3.
C  F1 AND F3 ARE MAPS FROM NITEM DIMENSIONAL EUCLIDEAN SPACE TO NITEM
C    DIMENSIONAL EUCLIDEAN SPACE; SO IS F2, EXCEPT IN THE ONE PARAMETER
C    MODEL, WHERE IT MAPS INTO THE REAL LINE.
C
C  MLE IS THE ITERATED FIT PROVIDED BY THE MODEL FOR THE
C    PROBABILITY OF A CORRECT RESPONSE.  IT IS AN NSUBJ X NITEM
C    ARRAY, AND IS REPRESENTED AS A FUNCTION OF THE ESTIMATED
C    PARAMETERS.
C  DER1 IS AN NSUBJ X NITEM MATRIX OF DERIVATIVES OF MLE WITH
C    RESPECT TO THE LOCATION PARAMETER. THAT IS, DER1(I,J) =
C    DMLE(I,J)/DD(J). SIMILARLY, DER2 AND DER3 ARE DERIVATIVES
C    OF MLE WITH RESPECT TO THE SLOPE AND ASYMPTOPE PARAMETERS.
C    TOGETHER, THEY DETERMINE THE GRADIENT OF THE LOG LIKELIHOOD
C    FUNCTION.
C  JACOB IS A VECTOR CONTAINING ALL NON-ZERO ENTRIES OF THE
C    JACOBIAN OF THE GRADIENT (THUS JACOB IS THE ESSENTIAL PART OF
C    THE HESSIAN OF THE LIKELIHOOD FUNCTION). IN THE 2 AND 3 PL,
C    THE JACOBIAN CONSISTS OF FLAG**2 SUBMATRICES, EACH NITEM X NITEM
C    AND DIAGONAL. MOREOVER, JACOB IS SYMMETRIC, AND THUS HAS ONLY
C    (FLAG**2+FLAG)/2 DISTINCT SUBMATRICES. RAPHSON PROGRESSES
C    THROUGH THESE DISTINCT SUBMATRICES IN NATURAL ORDER (E.G.,
C    UPPER LEFT, UPPER CENTER, UPPER RIGHT, ETC.); THUS,
```

```
C     THE FIRST NITEM ENTRIES ARE THE DIAGONAL ENTRIES OF THE THE
C     FIRST SUBMATRIX, THE SECOND NITEM ENTRIES FROM THE SECOND, ETC.
C     HOWEVER, IN THE CASE OF THE ONE PARAMETER MODEL, JACOB IS AN
C     NITEM+1 X NITEM+1 SYMMETRIC MATRIX WHOSE UPPER LEFT
C     NITEM X NITEM SUBMATRIX IS DIAGONAL. CONSEQUENTLY,
C     ITS INVERSION IS MORE DIFFICULT.
C   KEY INDICATES THE SUBMATRIX PRESENTLY BEING CONSIDERED.
C   K = J MOD NITEM
C   INVER CONTAINS THE DISTINCT ENTRIES OF THE INVERSE JACOBIAN,
C     ORDERED (IN VECTOR FORM) AS JACOB IS.
C   INV1 IS AN NITEM+1 X NITEM+1 MATRIX, REPRESENTING THE
C     INVERSE OF THE JACOBIAN IN THE ONE PARAMETER MODEL.
C   PSEUDO CONTAINS THE ENTRIES OF A DIAGONAL MATRIX AND IS
C     ANALAGOUS TO A DETERMINANT; IT IS USED TO CONDENSE THE
C     EXPRESSION OF THE ENTRIES OF INVER AND IS USED IN THE TWO
C     AND THREE PARAMETER MODELS.
C   DETERM IS THE DETERMINANT OF JACOB IN THE ONE PARAMETER MODEL.
C   DIPROD IS THE DETERMINANT OF THE UPPER LEFT SUBMATRIX OF JACOB;
C     IT IS THE PRODUCT OF THE FIRST NITEM DIAGONAL ENTRIES.
C   NORM IS A MEASURE OF THE ACCURACY OF THE ESTIMATED MODEL.
C   SUM1, SUM2, AND SUM3 ARE USED TO ADD UP THE CONTRIBUTIONS OF
C     INDIVIDUAL ADDENDS TO THE FUNCTION BEING ZEROED (THE GRADIENT).
C   I IS THE PRESENT ITEM.
C   J IS THE PRESENT "ABILITY".
C   KOUNT IS THE NUMBER OF ITERATIONS THUS FAR. IT IS NOT PERMITTED
C     TO EXCEED 100.
C
      INTEGER FLAG
      DIMENSION PROB(NROW,NITEM),U(NSUBJ),WT(NSUBJ)
      DIMENSION DER1(NROW,NITEM),DER2(NROW,NITEM),DER3(NROW,NITEM)
      DIMENSION C(NITEM),A(NITEM),D(NITEM),PSEUDO(NITEM)
      REAL JACOB((FLAG**2+FLAG)*NITEM/2),NORM
      REAL INVER((FLAG**2+FLAG)*NITEM/2),MLE(NROW,NITEM)
      REAL INV1(NITEM+1,NITEM+1)
      KOUNT=0
      GROLD=0.0
      GO TO (100,200,300),FLAG
100   A1OLD=A(1)
      DO 110 J=1,NITEM
        DO 160 I=1,NSUBJ
          IF (-A(1)*(U(I)-D(J)) .LT. 88.0) THEN
            MLE(I,J)=C(J)+(1-C(J))/(1+EXP(-A(1)*(U(I)-D(J))))
          ELSE
            MLE(I,J)=C(J)
          END IF
          DER1(I,J)=(MLE(I,J)-C(J))*(1-MLE(I,J))*(-A(1))/(1-C(J))
          DER2(I,J)=(MLE(I,J)-C(J))*(1-MLE(I,J))*(U(I)-D(J))/(1-C(J))
160     CONTINUE
110   CONTINUE
      DO 120 J=1,2*NITEM
        KEY=(J-1)/NITEM+1
        K=J-(KEY-1)*NITEM
        JACOB(J)=0.0
```

```
        DO 130 I=1,NSUBJ
          IF (KEY .EQ. 1) THEN
            JACOB(J)=JACOB(J)-(PROB(I,K)/MLE(I,K)**2+
     $        (1-PROB(I,K))/(1-MLE(I,K))**2)*DER1(I,K)**2*WT(I)
          ELSE IF (KEY .EQ. 2) THEN
            JACOB(J)=JACOB(J)-(PROB(I,K)/MLE(I,K)**2+
     $        (1-PROB(I,K))/(1-MLE(I,K))**2)*DER1(I,K)*DER2(I,K)*WT(I)
          END IF
130       CONTINUE
120     CONTINUE
        JACOB(2*NITEM+1)=0.0
        DO 135 J=1,NITEM
          DO 137 I=1,NSUBJ
            JACOB(2*NITEM+1)=JACOB(2*NITEM+1)-(PROB(I,K)/MLE(I,K)**2+
     $        (1-PROB(I,K))/(1-MLE(I,K))**2)*DER2(I,K)**2*WT(I)
137       CONTINUE
135     CONTINUE
        DO 172 M=1,2*NITEM+1
172     CONTINUE
        DIPROD=1.0
        DO 175 J=1,NITEM
          DIPROD=DIPROD*JACOB(J)
175     CONTINUE
        DETERM=JACOB(2*NITEM+1)*DIPROD
        DO 185 J=1,NITEM
          DETERM=DETERM-DIPROD*JACOB(J+NITEM)**2/JACOB(J)
185     CONTINUE
      DO 180 I=1,NITEM
      DO 190 J=1,NITEM
       INV1(I,J)=DIPROD*JACOB(I+NITEM)*JACOB(J+NITEM)
     &  /(JACOB(I)*JACOB(J)*DETERM)
190    CONTINUE
180    CONTINUE
      DO 181 I=1,NITEM
      INV1(I,I)=(DETERM+JACOB(I+NITEM)**2*DIPROD/JACOB(I))
     &  /(JACOB(I)*DETERM)
181   CONTINUE
      DO 183 I=1,NITEM
      INV1(I,NITEM+1)=-DIPROD*JACOB(I+NITEM)
     &  /(JACOB(I)*DETERM)
183    CONTINUE
       INV1(NITEM+1,NITEM+1)=DIPROD/DETERM
       DO 182 J=1,NITEM
         DO 192 I=J+1,NITEM+1
           INV1(I,J)=INV1(J,I)
192      CONTINUE
182    CONTINUE
       NORM=0.0
       SUM2=0.0
       DO 140 J=1,NITEM
         SUM1=0.0
         DO 150 I=1,NSUBJ
           SUM1=SUM1+DER1(I,J)*(PROB(I,J)/MLE(I,J)
```

```
     $              -(1-PROB(I,J))/(1-MLE(I,J)))*WT(I)
           SUM2=SUM2+DER2(I,J)*(PROB(I,J)/MLE(I,J)
     $              -(1-PROB(I,J))/(1-MLE(I,J)))*WT(I)
 150       CONTINUE
           DO 195 M=1,NITEM
             D(M)=D(M)-INV1(M,J)*SUM1
 195       CONTINUE
           A(1)=A(1)-INV1(NITEM+1,J)*SUM1
           NORM=NORM+SUM1**2
 140     CONTINUE
         DO 165 M=1,NITEM
           D(M)=D(M)-INV1(M,NITEM+1)*SUM2
 165     CONTINUE
         A(1)=A(1)-INV1(NITEM+1,NITEM+1)*SUM2
         IF(SUM2*GROLD.GE.0.0) GOTO 168
         A(1)=A(1)+0.5*INV1(NITEM+1,NITEM+1)*SUM2
         GROLD=0.0
         GOTO 169
 168     GROLD=SUM2
 169     CONTINUE
         NORM=NORM+SUM2**2
         NORM=SQRT(NORM)
         IF (NORM .LE. .001) GO TO 401
         IF (KOUNT .GE. 100) GO TO 402
         KOUNT=KOUNT+1
         GO TO 100
 200     DO 210 J=1,NITEM
           DO 270 I=1,NSUBJ
             IF (-A(J)*(U(I)-D(J)) .LT. 88.0) THEN
               MLE(I,J)=C(J)+(1-C(J))/(1+EXP(-A(J)*(U(I)-D(J))))
             ELSE
               MLE(I,J)=0.0
             END IF
             DER1(I,J)=(MLE(I,J)-C(J))*(1-MLE(I,J))*(-A(J))/(1-C(J))
             DER2(I,J)=(MLE(I,J)-C(J))*(1-MLE(I,J))*(U(I)-D(J))/(1-C(J))
 270       CONTINUE
 210     CONTINUE
         DO 220 J=1,3*NITEM
           JACOB(J)=0.0
           KEY=(J-1)/NITEM+1
           K=J-(KEY-1)*NITEM
           DO 230 I=1,NSUBJ
             IF (KEY .EQ. 1) THEN
               JACOB(J)=JACOB(J)-(PROB(I,K)/MLE(I,K)**2+
     $           (1-PROB(I,K))/(1-MLE(I,K))**2)*DER1(I,K)**2*WT(I)
             ELSE IF (KEY .EQ. 2) THEN
               JACOB(J)=JACOB(J)-(PROB(I,K)/MLE(I,K)**2+
     $           (1-PROB(I,K))/(1-MLE(I,K))**2)*DER1(I,K)*DER2(I,K)*WT(I)
             ELSE IF (KEY .EQ. 3) THEN
               JACOB(J)=JACOB(J)-(PROB(I,K)/MLE(I,K)**2+
     $           (1-PROB(I,K))/(1-MLE(I,K))**2)*DER2(I,K)**2*WT(I)
             END IF
 230       CONTINUE
```

```
220     CONTINUE
        DO 280 J=1,NITEM
          PSEUDO(J)=JACOB(J)*JACOB(J+2*NITEM)-JACOB(J+NITEM)**2
280     CONTINUE
        DO 240 J=1,3*NITEM
          KEY=(J-1)/NITEM+1
          IF (KEY .EQ. 1) THEN
            INVER(J)=JACOB(J+2*NITEM)/PSEUDO(J)
          ELSE IF (KEY .EQ. 2) THEN
            INVER(J)=-JACOB(J)/PSEUDO(J-NITEM)
          ELSE IF (KEY .EQ. 3) THEN
            INVER(J)=JACOB(J-2*NITEM)/PSEUDO(J-2*NITEM)
          END IF
240     CONTINUE
        NORM=0.0
        DO 250 J=1,NITEM
          SUM1=0.0
          SUM2=0.0
          DO 260 I=1,NSUBJ
            SUM1=SUM1+DER1(I,J)*(PROB(I,J)/MLE(I,J)
     $               -(1-PROB(I,J))/(1-MLE(I,J)))*WT(I)
            SUM2=SUM2+DER2(I,J)*(PROB(I,J)/MLE(I,J)
     $               -(1-PROB(I,J))/(1-MLE(I,J)))*WT(I)
260       CONTINUE
          D(J)=D(J)-(INVER(J)*SUM1+INVER(J+NITEM)*SUM2)
          A(J)=A(J)-(INVER(J+NITEM)*SUM1+INVER(J+2*NITEM)*SUM2)
          NORM=NORM+SUM1**2+SUM2**2
250     CONTINUE
        NORM=SQRT(NORM)
        IF (NORM .LE. .001) GO TO 401
        IF (KOUNT .GE. 100) GO TO 402
        KOUNT=KOUNT+1
        GO TO 200
300     DO 310 J=1,NITEM
          DO 320 I=1,NSUBJ
            IF (-A(J)*(U(I)-D(J)) .LT. 88.0) THEN
              MLE(I,J)=C(J)+(1-C(J))/(1+EXP(-A(J)*(U(I)-D(J))))
            ELSE
              MLE(I,J)=0.0
            END IF
            DER1(I,J)=(MLE(I,J)-C(J))*(1-MLE(I,J))*(-A(J))/(1-C(J))
            DER2(I,J)=(MLE(I,J)-C(J))*(1-MLE(I,J))*(U(I)-D(J))/(1-C(J))
            DER3(I,J)=(1-MLE(I,J))/(1-C(J))
320       CONTINUE
310     CONTINUE
        DO 330 J=1,6*NITEM
          JACOB(J)=0.0
          KEY=(J-1)/NITEM+1
          K=J-(KEY-1)*NITEM
          DO 340 I=1,NSUBJ
            IF (KEY .EQ. 1) THEN
              JACOB(J)=JACOB(J)-(PROB(I,K)/MLE(I,K)**2+
     $          (1-PROB(I,K))/(1-MLE(I,K))**2)*DER1(I,K)**2*WT(I)
```

```
            ELSE IF (KEY .EQ. 2) THEN
              JACOB(J)=JACOB(J)-(PROB(I,K)/MLE(I,K)**2+
     $          (1-PROB(I,K))/(1-MLE(I,K))**2)*DER1(I,K)*DER2(I,K)*WT(I)
            ELSE IF (KEY .EQ. 3) THEN
              JACOB(J)=JACOB(J)-(PROB(I,K)/MLE(I,K)**2+
     $          (1-PROB(I,K))/(1-MLE(I,K))**2)*DER1(I,K)*DER3(I,K)*WT(I)
            ELSE IF (KEY .EQ. 4) THEN
              JACOB(J)=JACOB(J)-(PROB(I,K)/MLE(I,K)**2+
     $          (1-PROB(I,K))/(1-MLE(I,K))**2)*DER2(I,K)**2*WT(I)
            ELSE IF (KEY .EQ. 5) THEN
              JACOB(J)=JACOB(J)-(PROB(I,K)/MLE(I,K)**2+
     $          (1-PROB(I,K))/(1-MLE(I,K))**2)*DER2(I,K)*DER3(I,K)*WT(I)
            ELSE IF (KEY .EQ. 6) THEN
              JACOB(J)=JACOB(J)-(PROB(I,K)/MLE(I,K)**2+
     $          (1-PROB(I,K))/(1-MLE(I,K))**2)*DER3(I,K)**2*WT(I)
            END IF
340       CONTINUE
330     CONTINUE
        DO 360 J=1,NITEM
          PSEUDO(J)=JACOB(J)*JACOB(J+3*NITEM)*JACOB(J+5*NITEM)
     $      +2.0*JACOB(J+NITEM)*JACOB(J+4*NITEM)*JACOB(J+2*NITEM)
     $      -JACOB(J)*JACOB(J+4*NITEM)**2
     $      -JACOB(J+NITEM)**2*JACOB(J+5*NITEM)
     $      -JACOB(J+2*NITEM)**2*JACOB(J+3*NITEM)
360     CONTINUE
        NORM=0.0
        DO 350 J=1,6*NITEM
          KEY=(J-1)/NITEM+1
          IF (KEY .EQ. 1) THEN
            INVER(J)=(JACOB(J+3*NITEM)*JACOB(J+5*NITEM)
     $        -JACOB(J+4*NITEM)**2)/PSEUDO(J)
          ELSE IF (KEY .EQ. 2) THEN
            INVER(J)=(JACOB(J+NITEM)*JACOB(J+3*NITEM)
     $        -JACOB(J)*JACOB(J+4*NITEM))/PSEUDO(J-NITEM)
          ELSE IF (KEY .EQ. 3) THEN
            INVER(J)=(JACOB(J-NITEM)*JACOB(J+2*NITEM)
     $        -JACOB(J)*JACOB(J+NITEM))/PSEUDO(J-2*NITEM)
          ELSE IF (KEY .EQ. 4) THEN
            INVER(J)=(JACOB(J-3*NITEM)*JACOB(J+2*NITEM)
     $        -JACOB(J-NITEM)**2)/PSEUDO(J-3*NITEM)
          ELSE IF (KEY .EQ. 5) THEN
            INVER(J)=(JACOB(J-3*NITEM)*JACOB(J-2*NITEM)
     $        -JACOB(J-4*NITEM)*JACOB(J))/PSEUDO(J-4*NITEM)
          ELSE IF (KEY .EQ. 6) THEN
            INVER(J)=(JACOB(J-5*NITEM)*JACOB(J-2*NITEM)
     $        -JACOB(J-4*NITEM)**2)/PSEUDO(J-5*NITEM)
          END IF
350     CONTINUE
        NORM=0.0
        DO 370 J=1,NITEM
          SUM1=0.0
          SUM2=0.0
          SUM3=0.0
```

```
      DO 380 I=1,NSUBJ
        SUM1=SUM1+DER1(I,J)*(PROB(I,J)/MLE(I,J)
     $             -(1-PROB(I,J))/(1-MLE(I,J)))*WT(I)
        SUM2=SUM2+DER2(I,J)*(PROB(I,J)/MLE(I,J)
     $             -(1-PROB(I,J))/(1-MLE(I,J)))*WT(I)
        SUM3=SUM3+DER3(I,J)*(PROB(I,J)/MLE(I,J)
     $             -(1-PROB(I,J))/(1-MLE(I,J)))*WT(I)
380   CONTINUE
      CORR= -(INVER(J)*SUM1+INVER(J+NITEM)*SUM2
     $              +INVER(J+2*NITEM)*SUM3)
      IF(ABS(CORR).GT.0.1) CORR=SIGN(0.1,CORR)
      D(J)=D(J)+CORR
      CORR= -(INVER(J+NITEM)*SUM1+INVER(J+3*NITEM)*SUM2
     $              +INVER(J+4*NITEM)*SUM3)
      IF(ABS(CORR).GT.0.1) CORR=SIGN(0.1,CORR)
      A(J)=A(J)+CORR
      CORR= -(INVER(J+2*NITEM)*SUM1+INVER(J+4*NITEM)*SUM2
     $              +INVER(J+5*NITEM)*SUM3)
      IF(ABS(CORR).GT.0.1) CORR=SIGN(0.1,CORR)
      C(J)=C(J)+CORR
        IF (C(J) .LE. 0.0) THEN
          C(J)=0.0
        ELSE IF (C(J) .GE. .5) THEN
          C(J)=.5
        END IF
        NORM=NORM+SUM1**2+SUM2**2+SUM3**2
370   CONTINUE
      NORM=SQRT(NORM)
      IF (NORM .LE. .001) GO TO 401
      IF (KOUNT .GE. 100) GO TO 402
      GO TO 300
402   WRITE(6,499)
499   FORMAT(' AFTER 100 ITERATIONS, RAPHSON DID NOT CONVERGE.')
      DO 498 J=1,NITEM
498   WRITE(6,497) D(J),A(J),C(J)
497   FORMAT(1X,'PARAMETER ESTIMATES AT FAILURE',3F10.3)
401   RETURN
      END
```

```
      SUBROUTINE AMINIT(NITEM,RESP,D,A,C,UHAT,SE,W)
C
C  AMJACK IS REALLY TWO
C      SUBROUTINES IN ONE. THE FIRST SECTION,
C    AMINIT, REPEATEDLY CALLS UPON MODE TO FILL OUT A VECTOR
C    OF MAXIMUM LIKELIHOOD ESTIMATES OF ABILITY FOR THE ENTIRE TEST,
C    AND A TABLE OF ANALOGOUS ENTRIES WHEN EACH ITEM IS SUCCESSIVELY
C    OMITTED. THE SECOND PORTION, AMJACK, USES THE TABLE TO CALCULATE
C    THE ROBUSTIFIED JACKKNIFE; PSEUDOVALUES ARE OBTAINED FROM THE
C    TABLE AND SORTED INTO TWO PILES, DEPENDING UPON WHETHER THE ITEM
C    CURRENTLY OMITTED WAS ANSWERED CORRECTLY OR NOT. THESE TWO PILES
C    ARE THEN USED TO GIVE TWO ESTIMATES OF LOCATION, USING A SINE
C    FUNCTION M-ESTIMATE (OR, IF NECESSARY, A MEDIAN), AND A WEIGHTED
C    AVERAGE IS TAKEN FROM THEM.
C
C  N.B. THE
C      SUBROUTINE AMT (A.K.A.
C      SUBROUTINE SH FROM THE PRINCETON
C    ROBUSTNESS STUDY) REQUIRES BOTH IR AND IW BE >= 4.
C
C  NITEM IS THE NUMBER OF ITEMS. AS THE DIMENSION STATEMENTS NOW
C    STAND, IT MUST BE <= 100.
C  RESP IS THE NITEM LONG ARRAY OF RESPONSES (1=CORRECT,0=NOT).
C  R IS THE NITEM LONG ARRAY OF CONVENIENT RESPONSES. AMJACK IS
C    ONLY USED WITH THE ONE PARAMETER LOGISTIC (RASCH) MODEL, FOR
C    WHICH THE RAW SCORE ALONE DETERMINES THE MAXIMUM LIKELIHOOD
C    ESTIMATE OF LOCATION.
C  RAWSC IS THE RAW SCORE. IT MUST BE BETWEEN 2 AND NITEM-2, INCLUSIVE,
C    FOR A DIFFUSE PRIOR.
C  D,A, AND C ARE THE PARAMETERS FOR THE LOGISTIC MODEL. EACH IS AN
C    NITEM LONG ARRAY, AND IN THE RASCH MODEL, ALL THE C ENTRIES ARE
C    ZERO AND ALL THE A ENTRIES ARE IDENTICAL.
C  D2 IS THE SAME AS D, EXCEPT THAT THE CURRENT ITEM IS OMITTED. THUS,
C    D2 IS AN NITEM-1 LONG ARRAY.
C  W IS THE ARRAY OF WORKING PARAMETERS. NONE ARE USED IN AMJACK.
C  RIGHT IS THE ARRAY OF PSEUDOVALUES FOR CORRECT RESPONSES.
C  WRONG IS THE ARRAY OF PSEUDOVALUES FOR INCORRECT RESPONSES.
C  IR INDEXES RIGHT, THEN INDICATES THE TOTAL NUMBER RIGHT.
C  IW IS AS IR, BUT FOR WRONG ANSWERS.
C  J INDEXES VECTOR AND THE ROWS OF TABLE.
C  K INDEXES THE COLUMNS OF TABLE.
C  VECTOR IS AN NITEM+1 ARRAY OF MLES OF ABILITY FOR THE ENTIRE TEST,
C    GIVEN THE RAW SCORE. VECTOR(J) IS THE ENTRY FOR A RAW SCORE OF J-1.
C  TABLE IS AN NITEM X NITEM ARRAY OF MLE'S OF ABILITY FOR A TEST
C    OMITTING ONE OF THE ITEMS. TABLE(J,K) IS THE MLE WITH THE KTH ITEM
C    OMITTED AND A RAW SCORE OF J-1.
C  TR IS THE ROBUSTIFIED ESTIMATE OF LOCATION FOR CORRECT RESPONSES.
C  TW IS AS TR, BUT FOR INCORRECT RESPONSES.
C  UHAT IS THE WEIGHTED AVERAGE OF TR AND TW.
C  SUM IS THE SUM OF THE SQUARED DEVIATIONS OF THE PSEUDOVALUES FROM
C    THE ESTIMATE.
C  SE IS THE STANDARD ERROR.
C
```

```
      INTEGER RESP(NITEM),R(100),RAWSC
      DIMENSION D(NITEM),A(NITEM),C(NITEM),W(10),D2(99)
      REAL*8 RIGHT(100),WRONG(100),TR,TW
      DIMENSION TABLE(100,100),VECTOR(101)
      DO 70 J=1,NITEM
        R(J)=0
70    CONTINUE
      CALL MODE(NITEM,R,D,A,C,VECTOR(1),SE,W)
      DO 71 J=1,NITEM
        R(J)=1
        CALL MODE(NITEM,R,D,A,C,VECTOR(J+1),SE,W)
71    CONTINUE
      DO 72 K=1,NITEM
        DO 73 J=1,NITEM-1
          IF (J .LT. K) THEN
            D2(J)=D(J)
          ELSE
            D2(J)=D(J+1)
          END IF
          R(J)=0
73      CONTINUE
        CALL MODE(NITEM-1,R,D2,A,C,TABLE(1,K),SE,W)
        DO 74 J=1,NITEM-1
          R(J)=1
          CALL MODE(NITEM-1,R,D2,A,C,TABLE(J+1,K),SE,W)
74      CONTINUE
72    CONTINUE
      RETURN
      ENTRY AMJACK(NITEM,RESP,D,A,C,UHAT,SE,W)
      RAWSC=0
      DO 75 J=1,NITEM
        RAWSC=RAWSC+RESP(J)
75    CONTINUE
      IR=0
      IW=0
      DO 76 K=1,NITEM
        IF (RESP(K) .EQ. 1) THEN
          IR=IR+1
          RIGHT(IR)=NITEM*VECTOR(RAWSC+1)-(NITEM-1)*TABLE(RAWSC,K)
        ELSE
          IW=IW+1
          WRONG(IW)=NITEM*VECTOR(RAWSC+1)-(NITEM-1)*TABLE(RAWSC+1,K)
        END IF
76    CONTINUE
      IF (IR .EQ. 0) THEN
        TR=0.0
      ELSE
        IF (IR .EQ. 1) THEN
          TR=RIGHT(1)
        ELSE
          IF (IR .EQ. 2) THEN
            TR=(RIGHT(1)+RIGHT(2))/2.0
          ELSE
```

```
            CALL SORT8(RIGHT,IR)
            IF (IR .EQ. 3) THEN
              TR=RIGHT(2)
            ELSE
              CALL AMT(RIGHT,IR,TR)
            END IF
          END IF
        END IF
      END IF
      IF (IW .EQ. 0) THEN
        TW=0.0
      ELSE
        IF (IW .EQ. 1) THEN
          TW=WRONG(1)
        ELSE
          IF (IW .EQ. 2) THEN
            TW=(WRONG(1)+WRONG(2))/2.0
          ELSE
            CALL SORT8(WRONG,IW)
            IF (IW .EQ. 3) THEN
              TW=WRONG(2)
            ELSE
              CALL AMT(WRONG,IW,TW)
            END IF
          END IF
        END IF
      END IF
      UHAT=(IR*TR+IW*TW)/NITEM
      SUM=0.0
      DO 77 J=1,IR
        SUM=SUM+(RIGHT(J)-UHAT)**2
77    CONTINUE
      DO 78 J=1,IW
        SUM=SUM+(WRONG(J)-UHAT)**2
78    CONTINUE
      SE=SQRT(SUM/NITEM/(NITEM-1))
      RETURN
      END
```

```
      SUBROUTINE AMT(X,N,T)
C
C  AMT IS COPIED FROM
C      SUBROUTINE SH OF THE PRINCETON ROBUSTNESS
C    STUDY. IT PROVIDES SINE FUNCTION M-ESTIMATES.
C
C  X IS THE INPUT ARRAY. IT ASSUMED SORTED.
C  N IS THE LENGTH OF ARRAY X. IT MUST BE AT LEAST 4.
C  T IS THE ESTIMATE.
C
        IMPLICIT REAL*8 (A-H,O-Z)
        REAL*8 X(N)
        DIMENSION DD(100)
        DATA PI /3.14159265357828/
        DATA ISTOP /20/
C          VECTOR ITER STORES A FREQUENCY CHART OF THE PERFORMANCE.
        INTEGER ITER(20)
        DATA ITER /20*0/
C          **COMPUTE INITIAL ESTIMATES S AND T**
        HUBERK=PI
        ML = (N+1)/2
        T = (X(ML)+X(N-ML+1))*0.5D0
        ND4 = N/4
        III=0
 166    III=III+1
        DO 90 I99=1,N
        D=X(I99)-T
        DD(I99)=DABS(D)
 90     CONTINUE
        CALL SORT8(DD,N)
        M1=(N+1)/2
        M2=(N+2)/2
        S=(DD(M1)+DD(M2))/2.0D0
        IF(S.LE.0.0) RETURN
        S=S*2.1D0
C          **COMPUTE WINSORIZING COUNTS AND DECIDE ON STOPPING**
        I = 0
        M1OLD = -1
        M3OLD = -1
 10     CONTINUE
        XL = T-S*HUBERK
        XU = T+S*HUBERK
        DO 11 J = 1,N
        IF (X(J) .GT. XL) GOTO 12
 11     CONTINUE
        J = N + 1
 12     M1 = J - 1
        DO 13 J = 1,N
        IF (X(N-J+1) .LT. XU) GOTO 14
 13     CONTINUE
        J = N + 1
 14     M3 = J - 1
        IF (M1 .EQ. M1OLD .AND. M3 .EQ. M3OLD) GOTO 50
```

```
      I = I + 1
      IF (I .GE. 20) GOTO 60
      FM1 = M1
      FM3 = M3
      FM2 = N - M3 - M1
      IF (FM2 .LT. 0.5D0) GOTO 60
      M1OLD = M1
      M3OLD = M3
C        **UPDATE ESTIMATE SET**
      M1P1 = M1 + 1
      NMM3 = N - M3
      SS=0.0D0
      SC=0.0D0
      DO 20 J = M1P1,NMM3
      Z=(X(J)-T)/S
      SZ=DSIN(Z)
      CZ=DCOS(Z)
      SS=SS+SZ
      SC=SC+CZ
 20   CONTINUE
      T=T+S*DATAN(SS/SC)
      GOTO 10
 50   ITER(I) = ITER(I) + 1
      IF (III .LT. 3) GOTO 166
      RETURN
 60   WRITE(6,1000) I,M1,M3,X
 1000 FORMAT('0ABORTING ANDREWS-MODIFIED-HUBER ON TRIAL',I4,
     &     '.  M1 AND M3 ARE',2I5,',AND THE SAMPLE IS:' / (8D16.7))
      RETURN
      END
```

```
      SUBROUTINE SORT8(V,N)
        IMPLICIT REAL*8 (A-H,O-Z)
        REAL*8 V(N)
C  SHELL SORT ALGORITHM CACM JULY 1964
C  TAKEN FROM PRINCETON ROBUSTNESS STUDY
        I=1
 1      I=I+I
        IF (I .LE. N) GO TO 1
        M=I-1
 2      M=M/2
        IF (M .EQ. 0) RETURN
        K=N-M
        DO 4 J=1,K
          L=J
 5        IF (L .LT. 1) GO TO 4
          IF (V(L+M) .GE. V(L)) GO TO 4
          X=V(L+M)
          V(L+M)=V(L)
          V(L)=X
          L=L-M
          GO TO 5
 4      CONTINUE
        GO TO 2
        END
```

```
      SUBROUTINE BIWT(NITEM,RESP,D,A,C,UHAT,SE,W)
C
C  BIWT ESTIMATES ABILITY BY FINDING THE ABILITY LEVEL AT WHICH THE
C    ITERATIVELY WEIGHTED PRODUCT OF THE LIKELIHOOD FUNCTION AND A
C    PRIOR DENSITY IS MAXIMIZED. IT USES A MODIFIED NEWTON'S METHOD
C    TO DO SO; IF THE GRADIENT CHANGES SIGN TWO CONSECUTIVE TIMES, THE
C    NEW ESTIMATE IS TAKEN TO BE ONLY HALF AS FAR FROM THE OLD ESTIMATE
C    AS THE SIMPLE NEWTON'S METHOD WOULD INDICATE.
C
C  NITEM IS THE NUMBER OF ITEMS.
C  RESP IS AN NITEM LONG ARRAY OF ITEM RESPONSES (1=CORRECT,0=NOT).
C  D,A, AND C ARE NITEM LONG ARRAYS OF LOGISTIC MODEL PARAMETERS.
C  UHAT IS THE ESTIMATED "ABILITY". IT STARTS AT WHATEVER VALUE IS
C    READ AND ENDS AT THE MODE.
C  SE IS THE STANDARD ERROR OF UHAT. IF IT IS UNDEFINED, IT IS SET TO 0.
C  W IS AN ARRAY OF WORKING PARAMETERS. ONLY W(3) AND W(4) ARE USED
C    IN BIWT.
C  SPRD IS A MEASURE OF PERMISSABLE SPREAD. IT DETERMINES THE AMOUNT
C    OF TRIMMING DONE.
C  DEV IS THE DEVIATION OF THE ABILITY ESTIMATE FROM THE DIFFICULTY
C    LEVEL, NORMALIZED BY SPRD.
C  NDIST INDICATES WHETHER THE PRIOR IS DIFFUSE OR N(0,1).
C  BWT IS THE BIWEIGHT.
C  P IS THE PROBABILITY OF CORRECT RESPONSE GENERATED BY THE LOGISTIC
C    MODEL.
C  DPDU IS THE DERIVATIVE OF P WITH RESPECT TO U EVALUATED AT UHAT.
C  GRAD IS THE FIRST DERIVATIVE OF THE N(0,1)-WEIGHTED LOG LIKELIHOOD.
C  GROLD IS 0.0 IF A HALF STEP HAS JUST BEEN TAKEN, AND IS THE
C    PREVIOUS VALUE OF GRAD OTHERWISE.
C  HESS IS THE EXPECTED VALUE OF THE SECOND DERIVATIVE OF THE LOG
C    LIKELIHOOD EVALUATED AT UHAT.
C  ITER RECORDS THE NUMBER OF ITERATIONS. IT MAY NOT EXCEED 40,
C    AND IF IT REACHES 40, AN ERROR MESSAGE IS PRODUCED.
C
      INTEGER RESP(NITEM)
      DIMENSION D(NITEM),A(NITEM),C(NITEM)
      DIMENSION W(10)
      SPRD=W(3)
      NDIST=W(4)
      GROLD=0.0
      ALPHA=1.0
      RSTRT=UHAT
      DO 31 ITER=1,40
        GRAD=0.0
        HESS=0.0
        DO 30 J=1,NITEM
          P=C(J)+(1-C(J))/(1+EXP(-A(J)*(UHAT-D(J))))
          DPDU=(P-C(J))*(1-P)*A(J)/(1-C(J))
          DEV=(UHAT-D(J))*A(J)/SPRD
          IF (ABS(DEV) .LT. 1.0) THEN
            BWT=(1-DEV**2)**2
          ELSE
            BWT=0.0
```

```
            END IF
            GRAD=GRAD+(RESP(J)/P-(1-RESP(J))/(1-P))*DPDU*BWT
            HESS=HESS-(1/P+1/(1-P))*DPDU**2*BWT
 30       CONTINUE
          IF (NDIST .EQ. 0) THEN
            GRAD=GRAD-UHAT
            HESS=HESS-1.0
          END IF
          IF(W(4).NE.0) THEN
            IF((UHAT.GE.W(4).AND.GRAD.GT.-0.001).OR.
     &         (UHAT.LE.-W(4).AND.GRAD.LT.0.001)) THEN
              UHAT=SIGN(W(4),UHAT)
              GRAD=0.
            END IF
          END IF
          IF (ABS(GRAD) .LE. .001) GO TO 32
          UHAT=UHAT-ALPHA*GRAD/HESS
          IF (GRAD*GROLD .LT. 0.0) THEN
            UHAT=UHAT+0.5*ALPHA*GRAD/HESS
            GROLD=0.0
          ELSE
            GROLD=GRAD
          END IF
          IF(ABS(UHAT).GT.6.) THEN
            UHAT=RSTRT
            ALPHA=0.5*ALPHA
            GROLD=0.
          END IF
 31       CONTINUE
          WRITE(6,830) RESP
 830      FORMAT(' AFTER 40 ITERATIONS, THE BIWT MODE WAS NOT REACHED.'/
     & 1X,100I1)
 32       IF (HESS .NE. 0.0) THEN
            SE=SQRT(-1.0/HESS)
          ELSE
            SE=0.0
          END IF
          IF(SE.GT.9.9) SE=9.9
          RETURN
          END
```

```
      SUBROUTINE HEST(NITEM,RESP,D,A,C,UHAT,SE,W)
C
C  HEST ESTIMATES ABILITY BY FINDING THE ABILITY LEVEL AT WHICH
C    THE PRODUCT OF THE LIKELIHOOD FUNCTION, THE PRODUCT OF THE
C    PROBABILITIES OF CORRECT AND INCORRECT RESPONSE ADJUSTED FOR
C    ASYMPTOPIC BEHAVIOR AND RAISED TO SOME POWER, AND A PRIOR
C    DENSITY IS MAXIMIZED. IT USES A MODIFIED NEWTON'S METHOD TO
C    DO SO; IF THE GRADIENT CHANGES SIGN TWO CONSECUTIVE TIMES,
C    THE NEW ESTIMATE IS TAKEN TO BE ONLY HALF AS FAR FROM THE OLD
C    ESTIMATE AS THE SIMPLE NEWTON'S METHOD WOULD INDICATE.
C
C  NITEM IS THE NUMBER OF ITEMS.
C  RESP IS AN NITEM LONG ARRAY OF ITEM RESPONSES (1=CORRECT,0=NOT).
C  D,A, AND C ARE NITEM LONG ARRAYS OF LOGISTIC MODEL PARAMETERS.
C  UHAT IS THE ESTIMATED "ABILITY". IT STARTS AT WHATEVER VALUE IS
C    READ AND ENDS AT THE MODE.
C  SE IS THE STANDARD ERROR OF UHAT. IF IT IS UNDEFINED, IT IS SET TO 0.
C  W IS AN ARRAY OF WORKING PARAMETERS. ONLY W(2) AND W(4) ARE USED
C    IN HEST.
C  H IS THE H OF H-ESTIMATE FAME.
C  NDIST INDICATES WHETHER THE PRIOR IS DIFFUSE OR N(0,1).
C  P IS THE PROBABILITY OF CORRECT RESPONSE GENERATED BY THE LOGISTIC
C    MODEL.
C  DPDU IS THE DERIVATIVE OF P WITH RESPECT TO U EVALUATED AT UHAT.
C  GRAD IS THE FIRST DERIVATIVE OF THE N(0,1)-WEIGHTED LOG LIKELIHOOD.
C  GROLD IS 0.0 IF A HALF STEP HAS JUST BEEN TAKEN, AND IS THE
C    PREVIOUS VALUE OF GRAD OTHERWISE.
C  HESS IS THE EXPECTED VALUE OF THE SECOND DERIVATIVE OF THE LOG
C    LIKELIHOOD EVALUATED AT UHAT.
C  VARNUM IS THE NUMERATOR OF THE ESTIMATE OF THE VARIANCE OF UHAT.
C  ITER RECORDS THE NUMBER OF ITERATIONS. IT MAY NOT EXCEED 40,
C    AND IF IT REACHES 40, AN ERROR MESSAGE IS PRODUCED.
C
      INTEGER RESP(NITEM)
      DIMENSION D(NITEM),A(NITEM),C(NITEM)
      DIMENSION W(10)
      GROLD=0.0
      H=W(2)
      NDIST=W(4)
      ALPHA=1.0
      RSTRT=UHAT
      DO 81 ITER=1,40
        GRAD=0.0
        HESS=0.0
        VARNUM=0.0
        DO 80 J=1,NITEM
          P=C(J)+(1-C(J))/(1+EXP(-A(J)*(UHAT-D(J))))
          DPDU=(P-C(J))*(1-P)*A(J)/(1-C(J))
          GRAD=GRAD+DPDU*(RESP(J)-P)/P/(1-P)
     &              *((P-C(J))*(1-P)/(1-C(J))**2)**H
          HESS=HESS-DPDU**2/P/(1-P)
     &              *((P-C(J))*(1-P)/(1-C(J))**2)**H
          VARNUM=VARNUM+(DPDU*(RESP(J)-P)/P/(1-P))**2
```

```
     &                    *((P-C(J))*(1-P)/(1-C(J))**2)**(2*H)
 80      CONTINUE
         IF (NDIST .EQ. 0) THEN
           GRAD=GRAD-UHAT
           HESS=HESS-1.0
         END IF
         IF(W(4).NE.0) THEN
           IF((UHAT.GE.W(4).AND.GRAD.GT.-0.001).OR.
     &        (UHAT.LE.-W(4).AND.GRAD.LT.0.001)) THEN
             UHAT=SIGN(W(4),UHAT)
             GRAD=0.
           END IF
         END IF
         IF (ABS(GRAD) .LE. .001) GO TO 82
         UHAT=UHAT-ALPHA*GRAD/HESS
         IF (GRAD*GROLD .LT. 0.0) THEN
           UHAT=UHAT+0.5*ALPHA*GRAD/HESS
           GROLD=0.0
         ELSE
           GROLD=GRAD
         END IF
         IF(ABS(UHAT).GT.6.) THEN
           UHAT=RSTRT
           ALPHA=0.5*ALPHA
           GROLD=0.
         END IF
 81      CONTINUE
         WRITE(6,880) RESP
 880     FORMAT(' AFTER 40 ITERATIONS, THE HEST MODE WAS NOT REACHED.'/
     &   1X,100I1)
 82      IF (HESS .NE. 0.0) THEN
           SE=SQRT(VARNUM/HESS**2)
         ELSE
           SE=0.0
         END IF
         IF(SE.GT.9.9) SE=9.9
         RETURN
         END
```

```
      SUBROUTINE HMEAN(NITEM,RESP,D,A,C,UHAT,SE,W)
C
C  HMEAN ESTIMATES AN ABILITY LEVEL BY TAKING A WEIGHTED AVERAGE
C    OF UNIFORMLY SPACED ABILITIES, WITH THE MODIFIED LIKELIHOOD
C    OF THE ABILITY UNDER THE APPROPRIATE LOGISTIC MODEL USED AS
C    ITS WEIGHT. THE LIKELIHOOD IS MODIFIED BY RAISING EACH TERM
C    TO A POWER DETERMINED BY H AND THE GENERATED PROPORTION OF
C    CORRECT RESPONSES.
C
C  NITEM IS THE NUMBER OF ITEMS.
C  RESP IS AN NITEM LONG VECTOR OF RESPONSES (1=CORRECT,0=NOT).
C  D,A, AND C ARE NITEM LONG VECTORS OF LOGISTIC MODEL PARAMETERS.
C  P IS THE PROBABILITY OF CORRECT RESPONSE GENERATED BY THE
C    LOGISTIC MODEL.
C  PWR IS THE POWER TO WHICH THE TERM OF THE LIKELIHOOD IS RAISED.
C  UHAT IS THE ESTIMATED ABILITY LEVEL FROM THE PRESENT ITERATION.
C  UOLD IS THE ESTIMATED ABILITY LEVEL FROM THE LAST ITERATION.
C  SE IS THE STANDARD ERROR OF UHAT.
C  VAR IS THE STANDARD ESTIMATE OF THE VARIANCE OF UHAT.
C  W IS A VECTOR OF WORKING PARAMETERS. ONLY W(1), W(2), AND W(4)
C    ARE USED IN HMEAN.
C  NQUAD IS THE NUMBER OF QUADRATURE POINTS; IT MAY NOT BE GREATER
C    THAN 100.
C  H IS THE H PARAMETER OF K-ESTIMATION FAME.
C  NDIST INDICATES WHETHER THE PRIOR IS DIFFUSE OR N(0,1).
C  LK IS AN NQUAD LONG ARRAY OF WEIGHTS PROPORTIONAL TO THE PRODUCT OF
C    THE LIKELIHOODS AND A N(0,1) DENSITY.
C  UQ IS AN NQUAD LONG ARRAY OF QUADRATURE POINTS (I.E., ABILITY LEVELS).
C  SUMLK IS THE SUM OF THE WEIGHTS.
C  ITER IS THE NUMBER OF ITERATIONS PERFORMED.
C
      INTEGER RESP(NITEM)
      DIMENSION D(NITEM),A(NITEM),C(NITEM)
      DIMENSION W(10),UQ(100)
      REAL LK(100)
      UOLD=0.0
      NQUAD=W(1)
      H=W(2)
      NDIST=W(4)
      IF(NDIST.EQ.0) THEN
        QS=4.
        QS2=8.
      END IF
      IF(W(4).NE.0.) THEN
        QI=W(4)/NQUAD
        QS=W(4)-QI
        QS2=2.*QS
      END IF
      DO 95 ITER=1,20
        SUMLK=0.0
        DO 97 I=1,NQUAD
          UQ(I)=-QS+QS2*(I-1)/(NQUAD-1)
          IF (NDIST .EQ. 0) THEN
```

```
            LK(I)=10.0E35*EXP(-UQ(I)**2/2.0)
          ELSE
            LK(I)=10.0E35
          END IF
97      CONTINUE
        DO 90 J=1,NITEM
          P=C(J)+(1-C(J))/(1+EXP(-A(J)*(UHAT-D(J))))
          PWR=((P-C(J))*(1-P)/(1-C(J))**2)**H
          DO 91 I=1,NQUAD
            P=C(J)+(1-C(J))/(1+EXP(-A(J)*(UQ(I)-D(J))))
            LK(I)=LK(I)*(P**RESP(J)*(1-P)**(1-RESP(J)))**PWR
91        CONTINUE
90      CONTINUE
        DO 98 I=1,NQUAD
          SUMLK=SUMLK+LK(I)
98      CONTINUE
        DO 93 I=1,NQUAD
          LK(I)=LK(I)/SUMLK
93      CONTINUE
        UHAT=0.0
        DO 94 I=1,NQUAD
          UHAT=UHAT+LK(I)*UQ(I)
94      CONTINUE
        IF (ABS(UHAT-UOLD) .LE. .001) GO TO 96
        UOLD=UHAT
95    CONTINUE
      WRITE(6,890) ITER,RESP
890   FORMAT(' AFTER',I3,' ITERATIONS, HMEAN DID NOT CONVERGE.'/
     &  1X,100I1)
96    VAR=0.0
      DO 92 I=1,NQUAD
        VAR=VAR+LK(I)*(UQ(I)-UHAT)**2
92    CONTINUE
C  VAR=VAR-(8.0/(NQUAD-1))**2/12.0
      SE=SQRT(VAR)
      RETURN
      END
```

```
      SUBROUTINE MEAN(NITEM,RESP,D,A,C,UHAT,SE,W)
C
C  MEAN ESTIMATES AN ABILITY LEVEL BY TAKING A WEIGHTED AVERAGE
C    OF UNIFORMLY SPACED ABILITIES, WITH THE LIKELIHOOD OF THE
C    ABILITY UNDER THE APPROPRIATE LOGISTIC MODEL USED AS ITS WEIGHT.
C
C  NITEM IS THE NUMBER OF ITEMS.
C  RESP IS AN NITEM LONG VECTOR OF RESPONSES (1=CORRECT,0=NOT).
C  D,A, AND C ARE NITEM LONG VECTORS OF LOGISTIC MODEL PARAMETERS.
C  P IS THE PROBABILITY OF CORRECT RESPONSE GENERATED BY THE
C    LOGISTIC MODEL.
C  UHAT IS THE ESTIMATED ABILITY LEVEL.
C  SE IS THE STANDARD ERROR OF UHAT.
C  VAR IS THE STANDARD ESTIMATE OF THE VARIANCE OF UHAT.
C  W IS A VECTOR OF WORKING PARAMETERS. ONLY W(1) AND W(4) ARE USED
C    IN MEAN.
C  NQUAD IS THE NUMBER OF QUADRATURE POINTS; IT MAY NOT BE GREATER
C    THAN 100.
C  NDIST INDICATES WHETHER THE PRIOR IS DIFFUSE OR N(0,1).
C  LK IS AN NQUAD LONG ARRAY OF WEIGHTS PROPORTIONAL TO THE PRODUCT OF
C    THE LIKELIHOODS AND A N(0,1) DENSITY.
C  UQ IS AN NQUAD LONG ARRAY OF QUADRATURE POINTS (I.E., ABILITY LEVELS).
C  SUMLK IS THE SUM OF THE WEIGHTS.
C
      INTEGER RESP(NITEM)
      DIMENSION D(NITEM),A(NITEM),C(NITEM)
      DIMENSION W(10),UQ(100)
      REAL LK(100)
      NQUAD=W(1)
      NDIST=W(4)
      IF(NDIST.EQ.0) THEN
        QS=4.
        QS2=8.
      END IF
      IF(W(4).NE.0.) THEN
        QI=W(4)/NQUAD
        QS=W(4)-QI
        QS2=2.*QS
      END IF
      SUMLK=0.0
      DO 40 I=1,NQUAD
        UQ(I)=-QS+QS2*(I-1)/(NQUAD-1)
        IF (NDIST .EQ. 0) THEN
          LK(I)=10.0**35*EXP(-UQ(I)**2/2.0)
        ELSE
          LK(I)=10.0**35
        END IF
        DO 41 J=1,NITEM
          P=C(J)+(1-C(J))/(1+EXP(-A(J)*(UQ(I)-D(J))))
          LK(I)=LK(I)*P**RESP(J)*(1-P)**(1-RESP(J))
41      CONTINUE
        SUMLK=SUMLK+LK(I)
40    CONTINUE
```

```
      DO 43 I=1,NQUAD
        LK(I)=LK(I)/SUMLK
43    CONTINUE
      UHAT=0.0
      DO 44 I=1,NQUAD
        UHAT=UHAT+LK(I)*UQ(I)
44    CONTINUE
      VAR=0.0
      DO 42 I=1,NQUAD
        VAR=VAR+LK(I)*(UQ(I)-UHAT)**2
42    CONTINUE
C  VAR=VAR-(8.0/(NQUAD-1))**2/12.0
      SE=SQRT(VAR)
      RETURN
      END
```

```
      SUBROUTINE MODE(NITEM,RESP,D,A,C,UHAT,SE,W)
C
C  MODE ESTIMATES ABILITY BY FINDING THE ABILITY LEVEL AT WHICH
C    THE PRODUCT OF THE LIKELIHOOD FUNCTION AND A PRIOR DENSITY
C    IS MAXIMIZED. IT USES  A MODIFIED NEWTON'S METHOD TO DO SO;
C    IF THE GRADIENT CHANGES SIGN TWO CONSECUTIVE TIMES, THE NEW
C    ESTIMATE IS TAKEN TO BE ONLY HALF AS FAR FROM THE OLD ESTIMATE
C    AS THE SIMPLE NEWTON'S METHOD WOULD INDICATE.
C
C  NITEM IS THE NUMBER OF ITEMS.
C  RESP IS AN NITEM LONG ARRAY OF ITEM RESPONSES (1=CORRECT,0=NOT).
C  D,A, AND C ARE NITEM LONG ARRAYS OF LOGISTIC MODEL PARAMETERS.
C  UHAT IS THE ESTIMATED "ABILITY". IT STARTS AT WHATEVER VALUE IS
C    READ AND ENDS AT THE MODE.
C  SE IS THE STANDARD ERROR OF UHAT. IF IT IS UNDEFINED, IT IS SET TO 0.
C  W IS AN ARRAY OF WORKING PARAMETERS. ONLY W(4) IS USED IN MODE.
C  NDIST INDICATES WHETHER THE PRIOR IS DIFFUSE OR N(0,1).
C  P IS THE PROBABILITY OF CORRECT RESPONSE GENERATED BY THE LOGISTIC
C    MODEL.
C  DPDU IS THE DERIVATIVE OF P WITH RESPECT TO U EVALUATED AT UHAT.
C  GRAD IS THE FIRST DERIVATIVE OF THE N(0,1)-WEIGHTED LOG LIKELIHOOD.
C  GROLD IS 0.0 IF A HALF STEP HAS JUST BEEN TAKEN, AND IS THE
C    PREVIOUS VALUE OF GRAD OTHERWISE.
C  HESS IS THE EXPECTED VALUE OF THE SECOND DERIVATIVE OF THE LOG
C    LIKELIHOOD EVALUATED AT UHAT.
C  ITER RECORDS THE NUMBER OF ITERATIONS. IT MAY NOT EXCEED 40,
C    AND IF IT REACHES 40, AN ERROR MESSAGE IS PRODUCED.
C
      INTEGER RESP(NITEM)
      DIMENSION D(NITEM),A(NITEM),C(NITEM)
      DIMENSION W(10)
      NDIST=W(4)
      GROLD=0.0
      ALPHA=1.0
      RSTRT=UHAT
      DO 51 ITER=1,40
        GRAD=0.0
        HESS=0.0
        DO 50 J=1,NITEM
          P=C(J)+(1-C(J))/(1+EXP(-A(J)*(UHAT-D(J))))
          DPDU=(P-C(J))*(1-P)*A(J)/(1-C(J))
          GRAD=GRAD+(RESP(J)/P-(1-RESP(J))/(1-P))*DPDU
          HESS=HESS-(1/P+1/(1-P))*DPDU**2
50      CONTINUE
        IF (NDIST .EQ. 0) THEN
          GRAD=GRAD-UHAT
          HESS=HESS-1.0
        END IF
        IF(W(4).NE.0) THEN
          IF((UHAT.GE.W(4).AND.GRAD.GT.-0.001).OR.
     &       (UHAT.LE.-W(4).AND.GRAD.LT.0.001)) THEN
            UHAT=SIGN(W(4),UHAT)
            GRAD=0.
```

```
            END IF
          END IF
          IF (ABS(GRAD) .LE. .001) GO TO 52
          UHAT=UHAT-ALPHA*GRAD/HESS
          IF (GRAD*GROLD .LT. 0.0) THEN
            UHAT=UHAT+0.5*ALPHA*GRAD/HESS
            GROLD=0.0
          ELSE
            GROLD=GRAD
          END IF
          IF(ABS(UHAT).GT.6.) THEN
            UHAT=RSTRT
            ALPHA=0.5*ALPHA
            GROLD=0.
          END IF
51        CONTINUE
          WRITE(6,850) RESP
 850      FORMAT(' AFTER 40 ITERATIONS, THE MODE WAS NOT REACHED.'
         1X,100I1)
 52       IF (HESS .NE. 0.0) THEN
            SE=SQRT(-1.0/HESS)
          ELSE
            SE=0.0
          END IF
          IF(SE.GT.9.9) SE=9.9
          RETURN
          END
```

```
      SUBROUTINE BIN1(NEST)
      DIMENSION IU(10,20,5),IS(10,20,5),D(10),DMEAN(10,5),DSD(10,5),
     & SMEAN(10,5),SSD(10,5),UHAT(10),SE(10),SIM(10,5),VM(10,5)
      CHARACTER*16 NAME
      CHARACTER*8 ESTIM(10)
      DATA ESTIM/'MEAN','M.25','M.5','M1.0','MODE',
     &  'H0.5','H1.0','H2.0','BIWT','AMJK'/
      DO 10 I=1,10
      DO 10 K=1,5
      DMEAN(I,K)=0
      DSD(I,K)=0
      SMEAN(I,K)=0
      SSD(I,K)=0
      SIM(I,K)=0
      VM(I,K)=0
      DO 10 J=1,20
      IU(I,J,K)=0
      IS(I,J,K)=0
10    CONTINUE
      RETURN
C
C
      ENTRY BIN2(U,UHAT,SE,NEST)
      K=U+3
      DO 40 I=1,NEST
      D(I)=U-UHAT(I)
      DMEAN(I,K)=DMEAN(I,K)+D(I)
      DSD(I,K)=DSD(I,K)+D(I)**2
      SMEAN(I,K)=SMEAN(I,K)+SE(I)
      SSD(I,K)=SSD(I,K)+SE(I)**2
      IF(SE(I).GT.0.) SIM(I,K)=SIM(I,K)+1./(SE(I)**2)
      VM(I,K)=VM(I,K)+SE(I)**2
      J=11-5*D(I)
      IF(J.GT.20) J=20
      IF(J.LT.1) J=1
      IU(I,J,K)=IU(I,J,K)+1
      J=21-10*SE(I)
      IF(J.LT.1) J=1
      IF(J.GT.20) J=20
      IS(I,J,K)=IS(I,J,K)+1
40    CONTINUE
      RETURN
C
C
      ENTRY BIN3(NAME,NEST,N)
      DO 100 I=1,NEST
      WRITE(12,71) NAME,ESTIM(I)
71    FORMAT('1',A16,5X,'ESTIMATOR=',A8/'0',29X,'BINNED U-UHAT'/
     & '0','   BIN   ',4X,'U=-2.0',3X,'U=-1.0',3X,'U= 0.0',3X,'U= 1.0',
     & 3X,'U= 2.0'/)
      WRITE(12,73) (IU(I,1,K),K=1,5)
73    FORMAT('    END   ',5I9)
      XL=1.6
```

```
      XR=1.8
      DO 75 J=2,19
      WRITE(12,74) XL,XR,(IU(I,J,K),K=1,5)
74    FORMAT(1X,F4.1,'-',F4.1,5I9)
      XL=XL-.2
      XR=XR-.2
75    CONTINUE
      WRITE(12,73) (IU(I,20,K),K=1,5)
C
      DO 77 K=1,5
      DMEAN(I,K)=DMEAN(I,K)/N
      DSD(I,K)=(DSD(I,K)-N*DMEAN(I,K)**2)/(N-1)
      IF(DSD(I,K).GT.0.) DSD(I,K)=SQRT(DSD(I,K))
      SMEAN(I,K)=SMEAN(I,K)/N
      SSD(I,K)=(SSD(I,K)-N*SMEAN(I,K)**2)/(N-1)
      IF(SSD(I,K).GT.0.) SSD(I,K)=SQRT(SSD(I,K))
      SIM(I,K)=SIM(I,K)/N
      VM(I,K)=VM(I,K)/N
77    CONTINUE
      WRITE(12,78) (DMEAN(I,K),K=1,5),(DSD(I,K),K=1,5)
78    FORMAT('0','MEAN D',4X,5F9.3/1X,'STD. DEV. ',5F9.3)
      WRITE(14,82) (DMEAN(I,K),K=1,5),(DSD(I,K),K=1,5)
 82   FORMAT(5F10.5)
      WRITE(12,81)
81    FORMAT('0',24X,'BINNED ESTIMATED S.E.'/
     & '0',' BIN ',4X,'U=-2.0',3X,'U=-1.0',3X,'U= 0.0',3X,'U= 1.0',
     & 3X,'U= 2.0'/)
      WRITE(12,73) (IS(I,1,K),K=1,5)
      XL=1.9
      XR=1.8
      DO 85 J=2,20
      WRITE(12,[illegible] XL,XR,(IS(I,J,K),K=1,5)
      XL=XL-.1
      XR=XR-.1
85    CONTINUE
C
      WRITE(12,88) (SMEAN(I,K),K=1,5),(SSD(I,K),K=1,5)
88    FORMAT('0','MEAN S.E. ',5F9.3/1X,'STD. DEV. ',5F9.3)
      DO 91 K=1,5
      SE(K)=0.
91    IF(VM(I,K).GT.0.) SE(K)=SQRT(VM(I,K))
      WRITE(12,93) (SE(K),K=1,5)
93    FORMAT(1X,'RT AVG VAR',5F9.3)
      DO 95 K=1,5
      SE(K)=0.
95    IF(SIM(I,K).GT.0.) SE(K)=SQRT(1./SIM(I,K))
      WRITE(12,97) (SE(K),K=1,5)
97    FORMAT(1X,'RT INV INF',5F9.3)
100   CONTINUE
C
      RETURN
      END
```

Simulation Program - A-38

```
      SUBROUTINE WRIT15
C      SUBROUTINE TO PRODUCE FILE 15
C
C          15 IS INPUT TO GRAPHOUT AND THE TREE# SERIES
      DIMENSION DBAR(5),SD(5),RMSE(5),DBARR(5),SDR(5),RMSER(5),UHAT(5)
      DIMENSION DBAR2(5),SD2(5),RMSE2(5)
      REWIND 14
      DO 100 LOOP=1,28
      READ(14,3) DBAR,SD
 3    FORMAT(5F10.5)
C
      SUMSQ=0.0
      SUM=0.0
      SUMSSQ=0.0
      UBAR=0.0
      SUMCR=0.0
      DO 20 I=1,5
      UHAT(I)=(I-3)-DBAR(I)
      UBAR=UBAR+0.2*UHAT(I)
      SUM=SUM+UHAT(I)
      SUMSQ=SUMSQ+UHAT(I)**2
      SUMSSQ=SUMSSQ+SD(I)**2
      SUMCR=SUMCR+(I-3)*UHAT(I)
 20   CONTINUE
C
      BETA=SUMCR/(SUMSQ-5.*UBAR**2+SUMSSQ)
      ALPHA=0.0
      ALPH2=0.0
      BETA2=SQRT(10./(SUMSQ-5*UBAR**2+SUMSSQ))
      DO 30 I=1,5
      ALPH2=ALPH2-0.2*BETA2*UHAT(I)
 30   ALPHA=ALPHA - 0.2*BETA*UHAT(I)
      DO 40 I=1,5
      RMSE(I)=SQRT(DBAR(I)**2+SD(I)**2)
      DBAR2(I)=(I-3)-(BETA2*UHAT(I)+ALPH2)
      SD2(I)=SQRT(BETA2**2*SD(I)**2)
      RMSE2(I)=SQRT(DBAR2(I)**2+SD2(I)**2)
      DBARR(I)=(I-3)-(BETA*UHAT(I)+ALPHA)
      SDR(I)=SQRT(BETA**2*SD(I)**2)
 40   RMSER(I)=SQRT(DBARR(I)**2+SDR(I)**2)
      WRITE(15,50) DBAR,SD,RMSE,DBAR2,SD2,RMSE2,DBARR,SDR,RMSER
 50   FORMAT(5F10.4)
 100  CONTINUE
 711  STOP
      END
```

Appendix B

```
Robust Ability Estimation Project--      4 ASVAB WK
A test of    4 items.
 Means use 10. quadrature points.
Cut-off for Biweight is 4.00
Prior code is (value=rectangle boundary,0=normal) 0.
Lots=0
Number of simulated examinees=  100
Seed=      456789.
```

Simulation Program - B-2

Some of the basic stuff for this run- 4 ASVAB WK

```
Item  1
 ak  -1.076 -2.856  1.519  0.193  2.221
 ck   0.118 -3.798  2.859 -1.919  2.741
 dk          0.098  0.186  0.305  0.411

5 Ps 0.237  0.326  0.466  0.641  0.783

Item  2
 ak  -3.026 -0.296  1.199 -0.703  2.826
 ck   0.096  0.563 -1.612  0.874  0.078
 dk          0.170  0.239  0.139  0.452

5 Ps 0.440  0.344  0.242  0.852  0.988

Item  3
 ak  -0.747 -1.810 -0.069  0.687  1.938
 ck  -1.594 -3.201  0.203  2.040  2.553
 dk          0.170  0.239  0.139  0.452

5 Ps 0.112  0.265  0.588  0.844  0.952

Item  4
 ak  -1.608 -0.364 -0.221 -0.774  2.967
 ck   0.019 -1.269  0.356  0.379  0.514
 dk          0.098  0.186  0.305  0.411

5 Ps 0.298  0.207  0.357  0.939  0.998
```

Estimated 1-PL Parameters-- 4 ASVAB WK

Item	Location	Slope
1	0.096	1.041
2	0.120	1.041
3	-0.308	1.041
4	-0.016	1.041

Simulation Program - B-4

4 ASVAB WK Estimator=MEAN

Binned u-uhat

Bin End	u=-2.0	u=-1.0	u= 0.0	u= 1.0	u= 2.0
1.6- 1.8					1
1.4- 1.6				1	26
1.2- 1.4					
1.0- 1.2			15	9	
0.8- 1.0					73
0.6- 0.8					
0.4- 0.6			30	40	
0.2- 0.4					
0.0- 0.2		29	37		
-0.2- 0.0				50	
-0.4--0.2					
-0.6--0.4		46	14		
-0.8--0.6					
-1.0--0.8	32	21			
-1.2--1.0			4		
-1.4--1.2					
-1.6--1.4	47	2			
-1.8--1.6					
End	21	2			
Mean d	-1.413	-0.452	0.221	0.248	1.063
Std. Dev.	0.481	0.484	0.566	0.389	0.269

Binned Estimated s.e.

Bin End	u=-2.0	u=-1.0	u= 0.0	u= 1.0	u= 2.0
1.9- 1.8					
1.8- 1.7					
1.7- 1.6					
1.6- 1.5					
1.5- 1.4					
1.4- 1.3					
1.3- 1.2					
1.2- 1.1					
1.1- 1.0					
1.0- 0.9					
0.9- 0.8					
0.8- 0.7	100	100	100	100	100
0.7- 0.6					
0.6- 0.5					
0.5- 0.4					
0.4- 0.3					
0.3- 0.2					
0.2- 0.1					
0.1- 0.0					
Mean s.e.	0.734	0.733	0.728	0.742	0.750
Std. Dev.	0.017	0.017	0.015	0.018	0.015

4 ASVAB WK Estimator=M.25

Binned u-uhat

Bin End	u=-2.0	u=-1.0	u= 0.0	u= 1.0	u= 2.0
1.6- 1.8					1
1.4- 1.6				1	26
1.2- 1.4					
1.0- 1.2				9	73
0.8- 1.0			15		
0.6- 0.8					
0.4- 0.6			30	40	
0.2- 0.4					
0.0- 0.2			37	50	
-0.2- 0.0		29			
-0.4--0.2					
-0.6--0.4		46	14		
-0.8--0.6					
-1.0--0.8		21	4		
-1.2--1.0	32				
-1.4--1.2					
-1.6--1.4	47	2			
-1.8--1.6					
End	21	2			
Mean d	-1.528	-0.559	0.178	0.401	1.257
Std. Dev.	0.381	0.384	0.453	0.304	0.207

Binned Estimated s.e.

Bin End	u=-2.0	u=-1.0	u= 0.0	u= 1.0	u= 2.0
1.9- 1.8					
1.8- 1.7					
1.7- 1.6					
1.6- 1.5					
1.5- 1.4					
1.4- 1.3					
1.3- 1.2					
1.2- 1.1					
1.1- 1.0					
1.0- 0.9					
0.9- 0.8	34	31	19	50	73
0.8- 0.7	66	69	81	50	27
0.7- 0.6					
0.6- 0.5					
0.5- 0.4					
0.4- 0.3					
0.3- 0.2					
0.2- 0.1					
0.1- 0.0					
Mean s.e.	0.789	0.788	0.785	0.795	0.800
Std. Dev.	0.011	0.011	0.010	0.012	0.010

Simulation Program - B-6

4 ASVAB WK Estimator=M.5

Binned u-uhat

Bin End	u=-2.0	u=-1.0	u= 0.0	u= 1.0	u= 2.0
1.6- 1.8					1
1.4- 1.6					26
1.2- 1.4				1	73
1.0- 1.2				9	
0.8- 1.0					
0.6- 0.8			15	40	
0.4- 0.6					
0.2- 0.4			30	50	
0.0- 0.2			37		
-0.2- 0.0					
-0.4--0.2		29	14		
-0.6--0.4					
-0.8--0.6		46	4		
-1.0--0.8		21			
-1.2--1.0					
-1.4--1.2	32	2			
-1.6--1.4					
-1.8--1.6	47	2			
End	21				
Mean d	-1.626	-0.650	0.141	0.528	1.416
Std. Dev.	0.300	0.302	0.358	0.238	0.160

Binned Estimated s.e.

Bin End	u=-2.0	u=-1.0	u= 0.0	u= 1.0	u= 2.0
1.9- 1.8					
1.8- 1.7					
1.7- 1.6					
1.6- 1.5					
1.5- 1.4					
1.4- 1.3					
1.3- 1.2					
1.2- 1.1					
1.1- 1.0					
1.0- 0.9					
0.9- 0.8	100	100	100	100	100
0.8- 0.7					
0.7- 0.6					
0.6- 0.5					
0.5- 0.4					
0.4- 0.3					
0.3- 0.2					
0.2- 0.1					
0.1- 0.0					
Mean s.e.	0.836	0.835	0.833	0.840	0.843
Std. Dev.	0.007	0.007	0.007	0.008	0.006

4 ASVAB WK Estimator=M1.0

Binned u-uhat

Bin End	u=-2.0	u=-1.0	u= 0.0	u= 1.0	u= 2.0
1.6- 1.8					1
1.4- 1.6					26
1.2- 1.4				1	73
1.0- 1.2				9	
0.8- 1.0					
0.6- 0.8				40	
0.4- 0.6			15	50	
0.2- 0.4			30		
0.0- 0.2			37		
-0.2- 0.0					
-0.4--0.2			14		
-0.6--0.4		29	4		
-0.8--0.6		46			
-1.0--0.8		21			
-1.2--1.0					
-1.4--1.2		2			
-1.6--1.4	32	2			
-1.8--1.6	47				
End	21				
Mean d	-1.778	-0.792	0.084	0.720	1.653
Std. Dev.	0.178	0.180	0.212	0.141	0.095

Binned Estimated s.e.

Bin End	u=-2.0	u=-1.0	u= 0.0	u= 1.0	u= 2.0
1.9- 1.8					
1.8- 1.7					
1.7- 1.6					
1.6- 1.5					
1.5- 1.4					
1.4- 1.3					
1.3- 1.2					
1.2- 1.1					
1.1- 1.0					
1.0- 0.9	100	100	100	100	100
0.9- 0.8					
0.8- 0.7					
0.7- 0.6					
0.6- 0.5					
0.5- 0.4					
0.4- 0.3					
0.3- 0.2					
0.2- 0.1					
0.1- 0.0					
Mean s.e.	0.906	0.906	0.905	0.908	0.909
Std. Dev.	0.002	0.002	0.002	0.002	0.002

Simulation Program - B-8

4 ASVAB WK Estimator=MODE

Binned u-uhat

Bin End	u=-2.0	u=-1.0	u= 0.0	u= 1.0	u= 2.0
1.6- 1.8					1
1.4- 1.6				1	26
1.2- 1.4					
1.0- 1.2			15	9	
0.8- 1.0					73
0.6- 0.8					
0.4- 0.6			30	40	
0.2- 0.4					
0.0- 0.2		29	37		
-0.2- 0.0				50	
-0.4--0.2					
-0.6--0.4		46	14		
-0.8--0.6					
-1.0--0.8	32	21			
-1.2--1.0			4		
-1.4--1.2					
-1.6--1.4	47	2			
-1.8--1.6					
End	21	2			
Mean d	-1.444	-0.480	0.210	0.290	1.114
Std. Dev.	0.456	0.459	0.536	0.369	0.256

Binned Estimated s.e.

Bin End	u=-2.0	u=-1.0	u= 0.0	u= 1.0	u= 2.0
1.9- 1.8					
1.8- 1.7					
1.7- 1.6					
1.6- 1.5					
1.5- 1.4					
1.4- 1.3					
1.3- 1.2					
1.2- 1.1					
1.1- 1.0					
1.0- 0.9					
0.9- 0.8					
0.8- 0.7	83	79	63	91	99
0.7- 0.6	17	21	37	9	1
0.6- 0.5					
0.5- 0.4					
0.4- 0.3					
0.3- 0.2					
0.2- 0.1					
0.1- 0.0					
Mean s.e.	0.716	0.715	0.709	0.724	0.734
Std. Dev.	0.019	0.019	0.017	0.020	0.017

4 ASVAB WK Estimator=H0.5

Binned u-uhat

Bin End	u=-2.0	u=-1.0	u= 0.0	u= 1.0	u= 2.0
1.6- 1.8					1
1.4- 1.6					26
1.2- 1.4				1	73
1.0- 1.2				9	
0.8- 1.0					
0.6- 0.8			15	40	
0.4- 0.6					
0.2- 0.4			30	50	
0.0- 0.2			37		
-0.2- 0.0					
-0.4--0.2		29	14		
-0.6--0.4					
-0.8--0.6		46	4		
-1.0--0.8		21			
-1.2--1.0					
-1.4--1.2	32	2			
-1.6--1.4					
-1.8--1.6	47	2			
End	21				
Mean d	-1.641	-0.664	0.136	0.549	1.441
Std. Dev.	0.287	0.289	0.342	0.228	0.154

Binned Estimated s.e.

Bin End	u=-2.0	u=-1.0	u= 0.0	u= 1.0	u= 2.0
1.9- 1.8					
1.8- 1.7					
1.7- 1.6					
1.6- 1.5					
1.5- 1.4					
1.4- 1.3					
1.3- 1.2					
1.2- 1.1					
1.1- 1.0					
1.0- 0.9					
0.9- 0.8					
0.8- 0.7					
0.7- 0.6					
0.6- 0.5					
0.5- 0.4					
0.4- 0.3	60	58	69	50	27
0.3- 0.2	40	42	31	50	73
0.2- 0.1					
0.1- 0.0					
Mean s.e.	0.294	0.295	0.300	0.269	0.249
Std. Dev.	0.047	0.046	0.036	0.044	0.036

Simulation Program - B-10

4 ASVAB WK Estimator=H1.0

Binned u-uhat

Bin End	u=-2.0	u=-1.0	u= 0.0	u= 1.0	u= 2.0
1.6- 1.8					24
1.4- 1.6					76
1.2- 1.4				1	
1.0- 1.2				9	
0.8- 1.0				33	
0.6- 0.8				57	
0.4- 0.6			15		
0.2- 0.4			30		
0.0- 0.2			37		
-0.2- 0.0			13		
-0.4--0.2			5		
-0.6--0.4		29			
-0.8--0.6		46			
-1.0--0.8		21			
-1.2--1.0		2			
-1.4--1.2		2			
-1.6--1.4	32				
-1.8--1.6	47				
End	21				
Mean d	-1.784	-0.798	0.083	0.729	1.664
Std. Dev.	0.173	0.174	0.206	0.137	0.093

Binned Estimated s.e.

Bin End	u=-2.0	u=-1.0	u= 0.0	u= 1.0	u= 2.0
1.9- 1.8					
1.8- 1.7					
1.7- 1.6					
1.6- 1.5					
1.5- 1.4					
1.4- 1.3					
1.3- 1.2					
1.2- 1.1					
1.1- 1.0					
1.0- 0.9					
0.9- 0.8					
0.8- 0.7					
0.7- 0.6					
0.6- 0.5					
0.5- 0.4					
0.4- 0.3					
0.3- 0.2	38	39	20	10	3
0.2- 0.1	62	61	80	90	97
0.1- 0.0					
Mean s.e.	0.188	0.187	0.187	0.175	0.167
Std. Dev.	0.020	0.019	0.015	0.017	0.014

4 ASVAB WK Estimator=H2.0

Binned u-uhat

Bin End	u=-2.0	u=-1.0	u= 0.0	u= 1.0	u= 2.0
1.6- 1.8					100
1.4- 1.6					
1.2- 1.4					
1.0- 1.2				9	
0.8- 1.0				91	
0.6- 0.8					
0.4- 0.6					
0.2- 0.4					
0.0- 0.2			79		
-0.2- 0.0			21		
-0.4--0.2					
-0.6--0.4					
-0.8--0.6					
-1.0--0.8		92			
-1.2--1.0		8			
-1.4--1.2					
-1.6--1.4					
-1.8--1.6					
End	100				
Mean d	-1.936	-0.940	0.025	0.918	1.899
Std. Dev.	0.052	0.052	0.062	0.042	0.028

Binned Estimated s.e.

Bin End	u=-2.0	u=-1.0	u= 0.0	u= 1.0	u= 2.0
1.9- 1.8					
1.8- 1.7					
1.7- 1.6					
1.6- 1.5					
1.5- 1.4					
1.4- 1.3					
1.3- 1.2					
1.2- 1.1					
1.1- 1.0					
1.0- 0.9					
0.9- 0.8					
0.8- 0.7					
0.7- 0.6					
0.6- 0.5					
0.5- 0.4					
0.4- 0.3					
0.3- 0.2					
0.2- 0.1					
0.1- 0.0	100	100	100	100	100
Mean s.e.	0.059	0.059	0.058	0.057	0.056
Std. Dev.	0.003	0.003	0.002	0.002	0.001

Simulation Program - B-12

4 ASVAB WK Estimator=BIWT

Binned u-uhat

Bin End	u=-2.0	u=-1.0	u= 0.0	u= 1.0	u= 2.0
1.6- 1.8					1
1.4- 1.6				1	26
1.2- 1.4					
1.0- 1.2				9	73
0.8- 1.0			15		
0.6- 0.8					
0.4- 0.6			30	40	
0.2- 0.4					
0.0- 0.2			37	50	
-0.2- 0.0		29			
-0.4--0.2					
-0.6--0.4		46	14		
-0.8--0.6					
-1.0--0.8		21	4		
-1.2--1.0	32				
-1.4--1.2					
-1.6--1.4	47	2			
-1.8--1.6					
End	21	2			
Mean d	-1.468	-0.503	0.202	0.332	0.172
Std. Dev.	0.425	0.428	0.507	0.338	0.229

Binned Estimated s.e.

Bin End	u=-2.0	u=-1.0	u= 0.0	u= 1.0	u= 2.0
1.9- 1.8					
1.8- 1.7					
1.7- 1.6					
1.6- 1.5					
1.5- 1.4					
1.4- 1.3					
1.3- 1.2					
1.2- 1.1					
1.1- 1.0					
1.0- 0.9					
0.9- 0.8					
0.8- 0.7	83	79	63	91	99
0.7- 0.6	17	21	37	9	1
0.6- 0.5					
0.5- 0.4					
0.4- 0.3					
0.3- 0.2					
0.2- 0.1					
0.1- 0.0					
Mean s.e.	0.724	0.722	0.714	0.735	0.747
Std. Dev.	0.024	0.024	0.022	0.026	0.021

4 ASVAB WK Estimator=AMJK

Binned u-uhat

Bin End	u=-2.0	u=-1.0	u= 0.0	u= 1.0	u= 2.0
1.6- 1.8				1	
1.4- 1.6			15		
1.2- 1.4					26
1.0- 1.2				9	
0.8- 1.0					
0.6- 0.8			30		
0.4- 0.6		29			73
0.2- 0.4				40	
0.0- 0.2			37		
-0.2- 0.0					
-0.4--0.2		46			
-0.6--0.4	32			50	
-0.8--0.6			14		
-1.0--0.8		21			
-1.2--1.0					
-1.4--1.2	47				
-1.6--1.4			4		
-1.8--1.6		2			
End	21	2			
Mean d	-1.224	-0.274	0.288	-0.036	0.705
Std. Dev.	0.655	0.659	0.765	0.543	0.382

Binned Estimated s.e.

Bin End	u=-2.0	u=-1.0	u= 0.0	u= 1.0	u= 2.0
1.9- 1.8					
1.8- 1.7					
1.7- 1.6					
1.6- 1.5					
1.5- 1.4					
1.4- 1.3					
1.3- 1.2					
1.2- 1.1					
1.1- 1.0					
1.0- 0.9					
0.9- 0.8					
0.8- 0.7					
0.7- 0.6					
0.6- 0.5	12	11	4	2	
0.5- 0.4	54	58	77	48	27
0.4- 0.3					
0.3- 0.2					
0.2- 0.1					
0.1- 0.0	34	31	19	50	73
Mean s.e.	0.327	0.338	0.382	0.245	0.147
Std. Dev.	0.206	0.200	0.167	0.209	0.182

Simulation Program - B-14

Estimated 2-PL Parameters-- 4 ASVAB WK

Item	Location	Slope
1	0.139	0.638
2	0.133	0.892
3	-0.265	1.323
4	-0.012	1.511

4 ASVAB WK Estimator=MEAN

Binned u-uhat

Bin End	u=-2.0	u=-1.0	u= 0.0	u= 1.0	u= 2.0
1.6- 1.8				1	4
1.4- 1.6					1
1.2- 1.4					22
1.0- 1.2			15	1	
0.8- 1.0				1	73
0.6- 0.8			15	16	
0.4- 0.6			12	5	
0.2- 0.4			3	26	
0.0- 0.2		29	24		
-0.2- 0.0			1	50	
-0.4--0.2		28	16		
-0.6--0.4		11	8		
-0.8--0.6		12	2		
-1.0--0.8	32	13			
-1.2--1.0		2	4		
-1.4--1.2	25	3			
-1.6--1.4	6				
-1.8--1.6	22				
End	15	2			
Mean d	-1.402	-0.406	0.215	0.197	1.035
Std. Dev.	0.481	0.471	0.600	0.334	0.194

Binned Estimated s.e.

Bin End	u=-2.0	u=-1.0	u= 0.0	u= 1.0	u= 2.0
1.9- 1.8					
1.8- 1.7					
1.7- 1.6					
1.6- 1.5					
1.5- 1.4					
1.4- 1.3					
1.3- 1.2					
1.2- 1.1					
1.1- 1.0					
1.0- 0.9					
0.9- 0.8					
0.8- 0.7	61	59	45	89	99
0.7- 0.6	39	41	55	11	1
0.6- 0.5					
0.5- 0.4					
0.4- 0.3					
0.3- 0.2					
0.2- 0.1					
0.1- 0.0					
Mean s.e.	0.714	0.713	0.708	0.732	0.743
Std. Dev.	0.022	0.021	0.020	0.022	0.016

Simulation Program - B-16

4 ASVAB WK Estimator=M.25

Binned u-uhat

Bin End	u=-2.0	u=-1.0	u= 0.0	u= 1.0	u= 2.0
1.6- 1.8					4
1.4- 1.6				1	23
1.2- 1.4					
1.0- 1.2				1	73
0.8- 1.0			15	1	
0.6- 0.8			12	16	
0.4- 0.6			3	31	
0.2- 0.4			15		
0.0- 0.2			24	50	
-0.2- 0.0		29	1		
-0.4--0.2		15	16		
-0.6--0.4		13	10		
-0.8--0.6		23			
-1.0--0.8		13	4		
-1.2--1.0	32	2			
-1.4--1.2	6	3			
-1.6--1.4	19				
-1.8--1.6	28				
End	15	2			
Mean d	-1.515	-0.518	0.175	0.362	1.237
Std. Dev.	0.382	0.374	0.482	0.259	0.147

Binned Estimated s.e.

Bin End	u=-2.0	u=-1.0	u= 0.0	u= 1.0	u= 2.0
1.9- 1.8					
1.8- 1.7					
1.7- 1.6					
1.6- 1.5					
1.5- 1.4					
1.4- 1.3					
1.3- 1.2					
1.2- 1.1					
1.1- 1.0					
1.0- 0.9					
0.9- 0.8	2	2	4	50	73
0.8- 0.7	98	98	96	50	27
0.7- 0.6					
0.6- 0.5					
0.5- 0.4					
0.4- 0.3					
0.3- 0.2					
0.2- 0.1					
0.1- 0.0					
Mean s.e.	0.771	0.771	0.767	0.786	0.794
Std. Dev.	0.015	0.015	0.014	0.016	0.011

4 ASVAB WK Estimator=M.5

Binned u-uhat

Bin End	u=-2.0	u=-1.0	u= 0.0	u= 1.0	u= 2.0
1.6- 1.8					1
1.4- 1.6					4
1.2- 1.4				1	22
1.0- 1.2					73
0.8- 1.0				1	
0.6- 0.8			15	10	
0.4- 0.6			15	12	
0.2- 0.4			15	26	
0.0- 0.2			24	50	
-0.2- 0.0			16		
-0.4--0.2		29	9		
-0.6--0.4		28	2		
-0.8--0.6		23	4		
-1.0--0.8		13			
-1.2--1.0		5			
-1.4--1.2	32				
-1.6--1.4	25				
-1.8--1.6	28	2			
End	15				
Mean d	-1.612	-0.615	0.140	0.496	1.399
Std. Dev.	0.303	0.297	0.383	0.202	0.112

Binned Estimated s.e.

Bin End	u=-2.0	u=-1.0	u= 0.0	u= 1.0	u= 2.0
1.9- 1.8					
1.8- 1.7					
1.7- 1.6					
1.6- 1.5					
1.5- 1.4					
1.4- 1.3					
1.3- 1.2					
1.2- 1.1					
1.1- 1.0					
1.0- 0.9					
0.9- 0.8	100	100	100	100	100
0.8- 0.7					
0.7- 0.6					
0.6- 0.5					
0.5- 0.4					
0.4- 0.3					
0.3- 0.2					
0.2- 0.1					
0.1- 0.0					
Mean s.e.	0.821	0.820	0.819	0.832	0.837
Std. Dev.	0.010	0.010	0.009	0.011	0.007

Simulation Program - B-18

4 ASVAB WK Estimator=M1.0

Binned u-uhat

Bin End	u=-2.0	u=-1.0	u= 0.0	u= 1.0	u= 2.0
1.6- 1.8					4
1.4- 1.6					96
1.2- 1.4				1	
1.0- 1.2				1	
0.8- 1.0				17	
0.6- 0.8				1	
0.4- 0.6			15		
0.2- 0.4			15		
0.0- 0.2			39		
-0.2- 0.0			17		
-0.4--0.2			14		
-0.6--0.4		29			
-0.8--0.6		28			
-1.0--0.8		36			
-1.2--1.0		5			
-1.4--1.2		2			
-1.6--1.4	32				
-1.8--1.6	25				
End	43				
Mean d	-1.765	-0.766	0.086	0.698	1.640
Std. Dev.	0.184	0.180	0.232	0.120	0.066

Binned Estimated s.e.

Bin End	u=-2.0	u=-1.0	u= 0.0	u= 1.0	u= 2.0
1.9- 1.8					
1.8- 1.7					
1.7- 1.6					
1.6- 1.5					
1.5- 1.4					
1.4- 1.3					
1 3- 1.2					
1.2- 1.1					
1.1- 1.0					
1.0- 0.9	3	2	6	76	95
0.9- 0.8	97	98	94	24	5
0.8- 0.7					
0.7- 0.6					
0.6- 0.5					
0.5- 0.4					
0.4- 0.3					
0.3- 0.2					
0.2- 0.1					
0.1- 0.0					
Mean s.e.	0.896	0.896	0.896	0.902	0.904
Std. Dev.	0.003	0.003	0.003	0.004	0.003

4 ASVAB WK Estimator=MODE

Binned u-uhat

Bin End	u=-2.0	u=-1.0	u= 0.0	u= 1.0	u= 2.0
1.6- 1.8				1	4
1.4- 1.6					1
1.2- 1.4					22
1.0- 1.2			15	1	73
0.8- 1.0				1	
0.6- 0.8			15	16	
0.4- 0.6			12	5	
0.2- 0.4			3	26	
0.0- 0.2		29	24	50	
-0.2- 0.0			1		
-0.4--0.2		28	16		
-0.6--0.4		11	8		
-0.8--0.6		12	2		
-1.0--0.8	32	13	4		
-1.2--1.0		2			
-1.4--1.2	25	3			
-1.6--1.4	6				
-1.8--1.6	22				
End	15	2			
Mean d	-1.441	-0.444	0.205	0.261	1.108
Std. Dev.	0.447	0.437	0.555	0.314	0.184

Binned Estimated s.e.

Bin End	u=-2.0	u=-1.0	u= 0.0	u= 1.0	u= 2.0
1.9- 1.8					
1.8- 1.7					
1.7- 1.6					
1.6- 1.5					
1.5- 1.4					
1.4- 1.3					
1.3- 1.2					
1.2- 1.1					
1.1- 1.0					
1.0- 0.9					
0.9- 0.8					
0.8- 0.7	35	31	21	76	95
0.7- 0.6	65	69	79	24	5
0.6- 0.5					
0.5- 0.4					
0.4- 0.3					
0.3- 0.2					
0.2- 0.1					
0.1- 0.0					
Mean s.e.	0.690	0.689	0.683	0.713	0.727
Std. Dev.	0.027	0.026	0.024	0.027	0.019

Simulation Program - B-20

4 ASVAB WK Estimator=H0.5

Binned u-uhat

Bin End	u=-2.0	u=-1.0	u= 0.0	u= 1.0	u= 2.0
1.6- 1.8					1
1.4- 1.6					4
1.2- 1.4				1	22
1.0- 1.2					73
0.8- 1.0				1	
0.6- 0.8				10	
0.4- 0.6			15	12	
0.2- 0.4			15	26	
0.0- 0.2			15	50	
-0.2- 0.0			24		
-0.4--0.2			16		
-0.6--0.4		29	9		
-0.8--0.6		28	2		
-1.0--0.8		23	4		
-1.2--1.0		13			
-1.4--1.2	32	5			
-1.6--1.4	25				
-1.8--1.6	28	2			
End	15				
Mean d	-1.631	-0.634	0.137	0.530	1.437
Std. Dev.	0.285	0.279	0.360	0.192	0.108

Binned Estimated s.e.

Bin End	u=-2.0	u=-1.0	u= 0.0	u= 1.0	u= 2.0
1.9- 1.8					
1.8- 1.7					
1.7- 1.6					
1.6- 1.5					
1.5- 1.4					
1.4- 1.3					
1.3- 1.2					
1.2- 1.1					
1.1- 1.0					
1.0- 0.9					
0.9- 0.8					
0.8- 0.7					
0.7- 0.6					
0.6- 0.5					
0.5- 0.4					
0.4- 0.3	59	53	47	12	3
0.3- 0.2	41	47	53	88	97
0.2- 0.1					
0.1- 0.0					
Mean s.e.	0.294	0.289	0.292	0.249	0.228
Std. Dev.	0.054	0.049	0.041	0.047	0.030

4 ASVAB WK Estimator=H1.0

Binned u-uhat

Bin End	u=-2.0	u=-1.0	u= 0.0	u= 1.0	u= 2.0
1.6- 1.8					4
1.4- 1.6					96
1.2- 1.4				1	
1.0- 1.2				1	
0.8- 1.0				17	
0.6- 0.8				81	
0.4- 0.6			15		
0.2- 0.4			15		
0.0- 0.2			39		
-0.2- 0.0			17		
-0.4--0.2			14		
-0.6--0.4		29			
-0.8--0.6		28			
-1.0--0.8		36			
-1.2--1.0		5			
-1.4--1.2		2			
-1.6--1.4	32				
-1.8--1.6	25				
End	43				
Mean d	-1.772	-0.773	0.086	0.714	1.658
Std. Dev.	0.176	0.172	0.222	0.116	0.064

Binned Estimated s.e.

Bin End	u=-2.0	u=-1.0	u= 0.0	u= 1.0	u= 2.0
1.9- 1.8					
1.8- 1.7					
1.7- 1.6					
1.6- 1.5					
1.5- 1.4					
1.4- 1.3					
1.3- 1.2					
1.2- 1.1					
1.1- 1.0					
1.0- 0.9					
0.9- 0.8					
0.8- 0.7					
0.7- 0.6					
0.6- 0.5					
0.5- 0.4					
0.4- 0.3					
0.3- 0.2	53	42	35	12	3
0.2- 0.1	47	58	65	88	97
0.1- 0.0					
Mean s.e.	0.195	0.192	0.190	0.168	0.158
Std. Dev.	0.023	0.020	0.017	0.022	0.014

Simulation Program - B-22

4 ASVAB WK Estimator=H2.0

Binned u-uhat

Bin End	u=-2.0	u=-1.0	u= 0.0	u= 1.0	u= 2.0
1.6- 1.8					100
1.4- 1.6					
1.2- 1.4					
1.0- 1.2				2	
0.8- 1.0				98	
0.6- 0.8					
0.4- 0.6					
0.2- 0.4					
0.0- 0.2			69		
-0.2- 0.0			31		
-0.4--0.2					
-0.6--0.4					
-0.8--0.6					
-1.0--0.8		93			
-1.2--1.0		7			
-1.4--1.2					
-1.6--1.4					
-1.8--1.6					
End	100				
Mean d	-1.930	-0.930	0.027	0.911	1.893
Std. Dev.	0.055	0.053	0.069	0.036	0.020

Binned Estimated s.e.

Bin End	u=-2.0	u=-1.0	u= 0.0	u= 1.0	u= 2.0
1.9- 1.8					
1.8- 1.7					
1.7- 1.6					
1.6- 1.5					
1.5- 1.4					
1.4- 1.3					
1.3- 1.2					
1.2- 1.1					
1.1- 1.0					
1.0- 0.9					
0.9- 0.8					
0.8- 0.7					
0.7- 0.6					
0.6- 0.5					
0.5- 0.4					
0.4- 0.3					
0.3- 0.2					
0.2- 0.1					
0.1- 0.0	100	100	100	100	100
Mean s.e.	0.065	0.064	0.062	0.058	0.057
Std. Dev.	0.003	0.003	0.003	0.003	0.002

4 AoVAB WK Estimator=BIWT

Binned u-uhat

Bin End	u=-2.0	u=-1.0	u= 0.0	u= 1.0	u= 2.0
1.6- 1.8					4
1.4- 1.6				1	1
1.2- 1.4					22
1.0- 1.2				1	73
0.8- 1.0			15	1	
0.6- 0.8			12	16	
0.4- 0.6			3	5	
0.2- 0.4			15	26	
0.0- 0.2			24	50	
-0.2- 0.0		29	1		
-0.4--0.2		15	16		
-0.6--0.4		13	8		
-0.8--0.6		23	2		
-1.0--0.8		13	4		
-1.2--1.0	32	2			
-1.4--1.2	6	3			
-1.6--1.4	19				
-1.8--1.6	28				
End	15	2			
Mean d	-1.469	-0.473	0.196	0.317	1.180
Std. Dev.	0.412	0.403	0.519	0.282	0.161

Binned Estimated s.e.

Bin End	u=-2.0	u=-1.0	u= 0.0	u= 1.0	u= 2.0
1.9- 1.8					
1.8- 1.7					
1.7- 1.6					
1.6- 1.5					
1.5- 1.4					
1.4- 1.3					
1.3- 1.2					
1.2- 1.1					
1.1- 1.0					
1.0- 0.9					
0.9- 0.8					
0.8- 0.7	41	46	33	76	95
0.7- 0.6	59	54	67	24	5
0.6- 0.5					
0.5- 0.4					
0.4- 0.3					
0.3- 0.2					
0.2- 0.1					
0.1- 0.0					
Mean s.e.	0.700	0.699	0.691	0.731	0.748
Std. Dev.	0.034	0.033	0.032	0.035	0.025

Simulation Program - B-24

Estimated 3-PL Parameters-- 4 ASVAB WK

Item	Location	Slope	Lwr Asymp
1	0.567	0.785	0.128
2	0.685	4.237	0.287
3	-0.197	1.390	0.034
4	0.357	4.096	0.210

4 ASVAB WK Estimator=MEAN

Binned u-uhat

Bin End	u=-2.0	u=-1.0	u= 0.0	u= 1.0	u= 2.0
1.6- 1.8				1	1
1.4- 1.6					3
1.2- 1.4					
1.0- 1.2				2	22
0.8- 1.0			18	9	
0.6- 0.8			12	5	73
0.4- 0.6			3	7	
0.2- 0.4			12		
0.0- 0.2			25	26	
-0.2- 0.0		42	15		
-0.4--0.2		15	8	50	
-0.6--0.4		12	1		
-0.8--0.6		11			
-1.0--0.8		15	2		
-1.2--1.0	51	3			
-1.4--1.2	6		4		
-1.6--1.4	22				
-1.8--1.6	6				
End	15	2			
Mean d	-1.391	-0.418	0.239	0.116	0.902
Std. Dev.	0.417	0.391	0.521	0.431	0.220

Binned Estimated s.e.

Bin End	u=-2.0	u=-1.0	u= 0.0	u= 1.0	u= 2.0
1.9- 1.8					
1.8- 1.7					
1.7- 1.6					
1.6- 1.5					
1.5- 1.4					
1.4- 1.3					
1.3- 1.2					
1.2- 1.1					
1.1- 1.0					
1.0- 0.9	5	2	1	1	
0.9- 0.8	17	7	4	7	3
0.8- 0.7	27	22	8	28	22
0.7- 0.6	50	61	49	50	73
0.6- 0.5	1	8	30	9	1
0.5- 0.4			8	5	1
0.4- 0.3					
0.3- 0.2					
0.2- 0.1					
0.1- 0.0					
Mean s.e.	0.736	0.695	0.637	0.689	0.693
Std. Dev.	0.083	0.076	0.085	0.074	0.036

Simulation Program - B-26

4 ASVAB WK Estimator=M.25

Binned u-uhat

Bin End	u=-2.0	u=-1.0	u= 0.0	u= 1.0	u= 2.0
1.6- 1.8				1	4
1.4- 1.6					
1.2- 1.4				1	22
1.0- 1.2				1	73
0.8- 1.0				9	
0.6- 0.8			18	12	
0.4- 0.6			15		
0.2- 0.4			13	26	
0.0- 0.2			24	50	
-0.2- 0.0			15		
-0.4--0.2		42	9		
-0.6--0.4		27			
-0.8--0.6		13	2		
-1.0--0.8		13	4		
-1.2--1.0		3			
-1.4--1.2	51				
-1.6--1.4	28				
-1.8--1.6	11				
End	10	2			
Mean d	-1.509	-0.544	0.183	0.330	1.164
Std. Dev.	0.303	0.291	0.400	0.336	0.169

Binned Estimated s.e.

Bin End	u=-2.0	u=-1.0	u= 0.0	u= 1.0	u= 2.0
1.9- 1.8					
1.8- 1.7					
1.7- 1.6					
1.6- 1.5					
1.5- 1.4					
1.4- 1.3					
1.3- 1.2					
1.2- 1.1					
1.1- 1.0					
1.0- 0.9	6	2	2	8	3
0.9- 0.8	39	26	15	78	95
0.8- 0.7	48	53	33		
0.7- 0.6	7	19	50	14	2
0.6- 0.5					
0.5- 0.4					
0.4- 0.3					
0.3- 0.2					
0.2- 0.1					
0.1- 0.0					
Mean s.e.	0.792	0.761	0.721	0.804	0.816
Std. Dev.	0.059	0.059	0.070	0.061	0.027

4 ASVAB WK Estimator=M.5

Binned u-uhat

Bin End	u=-2.0	u=-1.0	u= 0.0	u= 1.0	u= 2.0
1.6- 1.8					4
1.4- 1.6				1	1
1.2- 1.4				1	22
1.0- 1.2				3	73
0.8- 1.0				1	
0.6- 0.8				5	
0.4- 0.6			18	26	
0.2- 0.4			28	50	
0.0- 0.2			27		
-0.2- 0.0			13		
-0.4--0.2			8		
-0.6--0.4		42	2		
-0.8--0.6		40	4		
-1.0--0.8		15			
-1.2--1.0		1			
-1.4--1.2					
-1.6--1.4	51				
-1.8--1.6	39	2			
End	10				
Mean d	-1.618	-0.652	0.133	0.487	1.355
Std. Dev.	0.227	0.223	0.310	0.269	0.144

Binned Estimated s.e.

Bin End	u=-2.0	u=-1.0	u= 0.0	u= 1.0	u= 2.0
1.9- 1.8					
1.8- 1.7					
1.7- 1.6					
1.6- 1.5					
1.5- 1.4					
1.4- 1.3					
1.3- 1.2					
1.2- 1.1					
1.1- 1.0					
1.0- 0.9	6	2	2	8	3
0.9- 0.8	83	75	42	78	95
0.8- 0.7	11	23	56	14	2
0.7- 0.6					
0.6- 0.5					
0.5- 0.4					
0.4- 0.3					
0.3- 0.2					
0.2- 0.1					
0.1- 0.0					
Mean s.e.	0.843	0.820	0.795	0.876	0.887
Std. Dev.	0.043	0.045	0.055	0.046	0.020

Simulation Program - B-28

4 ASVAB WK Estimator=M1.0

Binned u-uhat

Bin End	u=-2.0	u=-1.0	u= 0.0	u= 1.0	u= 2.0
1.6- 1.8					95
1.4- 1.6					
1.2- 1.4				1	
1.0- 1.2				9	
0.8- 1.0				14	
0.6- 0.8				76	
0.4- 0.6					
0.2- 0.4			33		
0.0- 0.2			38		
-0.2- 0.0			23		
-0.4--0.2			6		
-0.6--0.4					
-0.8--0.6		64			
-1.0--0.8		31			
-1.2--1.0		3			
-1.4--1.2		2			
-1.6--1.4					
-1.8--1.6	73				
End	27				
Mean d	-1.782	-0.807	0.064	0.740	1.666
Std. Dev.	0.123	0.128	0.178	0.155	0.100

Binned Estimated s.e.

Bin End	u=-2.0	u=-1.0	u= 0.0	u= 1.0	u= 2.0
1.9- 1.8					
1.8- 1.7					
1.7- 1.6					
1.6- 1.5					
1.5- 1.4					
1.4- 1.3					
1.3- 1.2					
1.2- 1.1					
1.1- 1.0					
1.0- 0.9	83	63	52	98	100
0.9- 0.8	17	37	48	2	
0.8- 0.7					
0.7- 0.6					
0.6- 0.5					
0.5- 0.4					
0.4- 0.3					
0.3- 0.2					
0.2- 0.1					
0.1- 0.0					
Mean s.e.	0.916	0.906	0.900	0.951	0.958
Std. Dev.	0.022	0.023	0.029	0.020	0.007

4 ASVAB WK Estimator=MODE

Binned u-uhat

Bin End	u=-2.0	u=-1.0	u= 0.0	u= 1.0	u= 2.0
1.6- 1.8				1	1
1.4- 1.6					1
1.2- 1.4					3
1.0- 1.2					22
0.8- 1.0					73
0.6- 0.8			18	10	
0.4- 0.6			12	5	
0.2- 0.4				8	
0.0- 0.2			33	26	
-0.2- 0.0			3	50	
-0.4--0.2		42	18		
-0.6--0.4		15	8		
-0.8--0.6		5	2		
-1.0--0.8		20	2		
-1.2--1.0		7	4		
-1.4--1.2	51	7			
-1.6--1.4	6				
-1.8--1.6	6	2			
End	37	2			
Mean d	-1.658	-0.655	0.028	0.120	0.988
Std. Dev.	0.470	0.411	0.480	0.290	0.120

Binned Estimated s.e.

Bin End	u=-2.0	u=-1.0	u= 0.0	u= 1.0	u= 2.0
1.9- 1.8					
1.8- 1.7					
1.7- 1.6					
1.6- 1.5					
1.5- 1.4					
1.4- 1.3					
1.3- 1.2					
1.2- 1.1					
1.1- 1.0					
1.0- 0.9					
0.9- 0.8	51	42	18	1	
0.8- 0.7	18	31	24		
0.7- 0.6	5	9	21		
0.6- 0.5	18	9	7	50	73
0.5- 0.4	8	9	30	49	27
0.4- 0.3					
0.3- 0.2					
0.2- 0.1					
0.1- 0.0					
Mean s.e.	0.723	0.728	0.626	0.487	0.504
Std. Dev.	0.134	0.123	0.151	0.056	0.036

Simulation Program - B-30

4 ASVAB WK Estimator=H0.5

Binned u-uhat

Bin End	u=-2.0	u=-1.0	u= 0.0	u= 1.0	u= 2.0
1.6- 1.8					1
1.4- 1.6				1	1
1.2- 1.4					25
1.0- 1.2					73
0.8- 1.0				2	
0.6- 0.8				8	
0.4- 0.6			18	6	
0.2- 0.4			15	33	
0.0- 0.2			18	50	
-0.2- 0.0			18		
-0.4--0.2			15		
-0.6--0.4		42	9		
-0.8--0.6		27	3		
-1.0--0.8		15	4		
-1.2--1.0		7			
-1.4--1.2		5			
-1.6--1.4	51	2			
-1.8--1.6	28				
End	21	2			
Mean d	-1.762	-0.785	-0.024	0.307	1.215
Std. Dev.	0.300	0.269	0.348	0.206	0.078

Binned Estimated s.e.

Bin End	u=-2.0	u=-1.0	u= 0.0	u= 1.0	u= 2.0
1.9- 1.8					
1.8- 1.7					
1.7- 1.6					
1.6- 1.5					
1.5- 1.4					
1.4- 1.3					
1.3- 1.2					
1.2- 1.1					
1.1- 1.0					
1.0- 0.9					
0.9- 0.8					
0.8- 0.7					
0.7- 0.6					
0.6- 0.5					
0.5- 0.4					
0.4- 0.3	17	11	6	1	
0.3- 0.2	81	87	90	49	27
0.2- 0.1	2	2	4	50	73
0.1- 0.0					
Mean s.e.	0.264	0.265	0.261	0.203	0.184
Std. Dev.	0.031	0.030	0.029	0.038	0.021

4 ASVAB WK Estimator=H1.0

Binned u-uhat

Bin End	u=-2.0	u=-1.0	u= 0.0	u= 1.0	u= 2.0
1.6- 1.8					5
1.4- 1.6					22
1.2- 1.4				1	73
1.0- 1.2				1	
0.8- 1.0				3	
0.6- 0.8				19	
0.4- 0.6				26	
0.2- 0.4			18	50	
0.0- 0.2			34		
-0.2- 0.0			21		
-0.4--0.2			21		
-0.6--0.4			2		
-0.8--0.6		42	4		
-1.0--0.8		44			
-1.2--1.0		11			
-1.4--1.2		1			
-1.6--1.4					
-1.8--1.6	51	2			
End	49				
Mean d	-1.848	-0.875	-0.042	0.506	1.435
Std. Dev.	0.164	0.162	0.236	0.156	0.065

Binned Estimated s.e.

Bin End	u=-2.0	u=-1.0	u= 0.0	u= 1.0	u= 2.0
1.9- 1.8					
1.8- 1.7					
1.7- 1.6					
1.6- 1.5					
1.5- 1.4					
1.4- 1.3					
1.3- 1.2					
1.2- 1.1					
1.1- 1.0					
1.0- 0.9					
0.9- 0.8					
0.8- 0.7					
0.7- 0.6					
0.6- 0.5					
0.5- 0.4					
0.4- 0.3					
0.3- 0.2	2	5	16	15	4
0.2- 0.1	98	95	84	85	96
0.1- 0.0					
Mean s.e.	0.166	0.166	0.173	0.182	0.175
Std. Dev.	0.018	0.022	0.025	0.017	0.010

Simulation Program - B-32

4 ASVAB WK Estimator=H2.0

Binned u-uhat

Bin End	u=-2.0	u=-1.0	u= 0.0	u= 1.0	u= 2.0
1.6- 1.8					100
1.4- 1.6					
1.2- 1.4					
1.0- 1.2				2	
0.8- 1.0				98	
0.6- 0.8					
0.4- 0.6					
0.2- 0.4					
0.0- 0.2			52		
-0.2- 0.0			48		
-0.4--0.2					
-0.6--0.4					
-0.8--0.6					
-1.0--0.8		86			
-1.2--1.0		14			
-1.4--1.2					
-1.6--1.4					
-1.8--1.6					
End	100				
Mean d	-1.953	-0.962	-0.007	0.919	1.904
Std. Dev.	0.036	0.041	0.057	0.038	0.025

Binned Estimated s.e.

Bin End	u=-2.0	u=-1.0	u= 0.0	u= 1.0	u= 2.0
1.9- 1.8					
1.8- 1.7					
1.7- 1.6					
1.6- 1.5					
1.5- 1.4					
1.4- 1.3					
1.3- 1.2					
1.2- 1.1					
1.1- 1.0					
1.0- 0.9					
0.9- 0.8					
0.8- 0.7					
0.7- 0.6					
0.6- 0.5					
0.5- 0.4					
0.4- 0.3					
0.3- 0.2					
0.2- 0.1					
0.1- 0.0	100	100	100	100	100
Mean s.e.	0.049	0.048	0.049	0.059	0.060
Std. Dev.	0.004	0.005	0.007	0.004	0.003

4 ASVAB WK Estimator=BIWT

Binned u-uhat

Bin End	u=-2.0	u=-1.0	u= 0.0	u= 1.0	u= 2.0
1.6- 1.8				2	
1.4- 1.6					2
1.2- 1.4					3
1.0- 1.2					95
0.8- 1.0				2	
0.6- 0.8			22	1	
0.4- 0.6				12	
0.2- 0.4			12	7	
0.0- 0.2			18	76	
-0.2- 0.0			18		
-0.4--0.2		51	3		
-0.6--0.4			20		
-0.8--0.6		20	1		
-1.0--0.8		15	6		
-1.2--1.0		7			
-1.4--1.2	72	4			
-1.6--1.4		1			
-1.8--1.6	12				
End	16	2			
Mean d	-1.522	-0.630	0.025	0.191	1.065
Std. Dev.	0.356	0.361	0.459	0.296	0.098

Binned Estimated s.e.

Bin End	u=-2.0	u=-1.0	u= 0.0	u= 1.0	u= 2.0
1.9- 1.8					
1.8- 1.7					
1.7- 1.6					
1.6- 1.5					
1.5- 1.4					
1.4- 1.3					
1.3- 1.2					
1.2- 1.1					
1.1- 1.0					
1.0- 0.9					
0.9- 0.8	72	51	22	2	
0.8- 0.7	18	31	24		
0.7- 0.6	5	9	21		
0.6- 0.5	2	4	7	52	73
0.5- 0.4	3	5	26	46	27
0.4- 0.3					
0.3- 0.2					
0.2- 0.1					
0.1- 0.0					
Mean s.e.	0.795	0.764	0.650	0.516	0.532
Std. Dev.	0.082	0.104	0.152	0.068	0.041

Appendix C

```
*#FRNH GRAPHOUT=(NWARN,FORM)
C          PROGRAM TO GRAPH ROBUST SIMULATION RESULTS
C          T(I,J,K), I=1,28; J=1,9; K=1,5
C           CONTAINS, FOR
C           I=I, STATISTICS FOR ESTIMATOR I
C           J-J, 1,9 DIFFERENT CRITERIA
C           K=K, 1,5, U=-2,-1,0,+1,+2
C           PROGRAM READS T FROM FILE "01"
C           AND PLOTS IN STANDARD FORMAT ON REQUEST
C
       LOGICAL DEV
       DIMENSION T(28,9,5), IMODEL(28),IESTIM(28),JCRIT(9),
     & U(5),IU(7),IY(3),Y(5),LU(16),ID(3)
C
      DATA IMODEL/'1-PL','1-PL','1-PL','1-PL','1-PL','1-PL','1-PL',
     & '1-PL','1-PL','1-PL','2-PL','2-PL','2-PL','2-PL','2-PL',
     & '2-PL','2-PL','2-PL','2-PL','3-PL','3-PL','3-PL','3-PL','3-PL',
     & '3-PL','3-PL','3-PL','3-PL'/
      DATA IESTIM/'MEAN','M.25','M.50','M1.0','MODE',
     & 'H0.5','H1.0','H2.0','BIWT','AMJK',
     & 'MEAN','M.25','M.50','M1.0','MODE',
     & 'H0.5','H1.0','H2.0','BIWT',
     & 'MEAN','M.25','M.50','M1.0','MODE',
     & 'H0.5','H1.0','H2.0','BIWT'/
      DATA JCRIT/'DBAR','S.D.','RMSE','DBVM','SDVM','REVM',
     & 'DBMI','SDMI','REMI'/
      DATA U/-2.,-1.,0.,1.,2./
      DATA NEW,ISTOP,D,V/'NEW','STOP','D','V'/
      DATA IU/' ',' -2 ',' -1 ','  0 ',' +1 ',' +2 ',' '/
      DATA LU/'U'/
      DATA IY/'   0','   1','   2'/
      DATA ID/'  -2','   0','  +2'/
      DATA IPLUS/'+'/
      WRITE(6,2)
 2    FORMAT(' PROGRAM TO PLOT ROBUST SIMULATION RESULTS')
      DO 10 I=1,28
      DO 9 J=1,9
      READ(15,5) (T(I,J,K),K=1,5)
 5    FORMAT(5F10.4)
 9    CONTINUE
 10   CONTINUE
C
 100  WRITE(6,101)
 101  FORMAT(' READY PLOTTER; ENTER "D" FOR BIAS, "V" FOR SE PLOTS')
      READ(5,102) AIT
 102  FORMAT(A1)
      DEV=.FALSE.
      IF(AIT.EQ.D) DEV=.TRUE.
      CALL START(IX1,IY1,IX2,IY2)
      CALL XAXIS11(1,7,7,1,LU,IU,IX1,IY1,IX2,IY2)
```

```
      IF(.NOT.DEV) CALL YAXIS85(1,3,3,0,LU,IY,IX1,IY1,IX2,IY2)
      IF(DEV) CALL YAXIS85(1,3,3,0,LU,ID,IX1,IY1,IX2,IY2)
C
 200  WRITE(6,201) (IMODEL(I),I=1,10),(IESTIM(I),I=1,10)
 201  FORMAT('0MODELS/ESTIMATORS AVAILABLE:'/
     & 1X,10(1X,A4)/1X,10(1X,A4))
      WRITE(6,202) (IMODEL(I),I=11,19),(IESTIM(I),I=11,19)
 202  FORMAT('0',9(1X,A4)/1X,9(1X,A4))
      WRITE(6,202) (IMODEL(I),I=20,28),(IESTIM(I),I=20,28)
      WRITE(6,203) (JCRIT(J),J=1,9)
 203  FORMAT('0CRITERIA:'/'0',9(1X,A4))
      WRITE(6,204)
 204  FORMAT('0ENTER MODEL, ESTIMATOR, AND CRITERION DESIRED'/
     & ' USE 4-CHARACTER CODES (UPPER CASE/; SEPARATE BY 1 SPACE'/
     & ' LINE TYPE AS AN INTEGER AT END; 0=SOLID')
      READ(5,205) IM,IE,JC,ITYPE
 205  FORMAT(A4,1X,A4,1X,A4,1X,I1)
      IF(IM.EQ.NEW) GOTO 100
      IF(IM.EQ.ISTOP) GOTO 711
      DO 210 I=1,28
      IF(IM.EQ.IMODEL(I).AND.IE.EQ.IESTIM(I)) GOTO 215
 210  CONTINUE
      WRITE(6,211) IM,IE
 211  FORMAT(2(1X,A4),' UNRECOGNIZED; TRY AGAIN')
      GOTO 200
 215  DO 220 J=1,9
      IF(JC.EQ.JCRIT(J)) GOTO 225
 220  CONTINUE
      WRITE(6,221) JC
 221  FORMAT(' ',A4,'UNRECOGNIZED; TRY AGAIN')
      GOTO 200
C
 225  CALL PLOTON
      CALL LINETP(ITYPE)
      DO 230 K=1,5
 230  Y(K)=T(I,J,K)
      IF(.NOT.DEV) CALL POINT(U,Y,-3,0,3,2,1,IPLUS,5,0.2)
      IF(DEV) CALL POINT(U,Y,-3,-2,3,2,1,IPLUS,5,0.2)
      CALL PLOTON
      CALL LINETP(0)
      CALL PLOTOF
      CALL TYPER
      GOTO 200
 711  STOP
      END
      SUBROUTINE START(IX1,IY1,IX2,IY2)
      CHARACTER*1 ESC/'{ESC}'/,CTLT/'{CTL-T}'/
      IX1=2000
      IY1=2000
      IX2=8000
      IY2=6000
      WRITE(6,4) ESC
 4    FORMAT(1X,' SETTING PLOTTER...',A1,'.(')
      WRITE(6,5) IX1,IY1,IX2,IY2
```

```
 5    FORMAT(1X,'IN;IP',4(I5,','),';DT{CTL-T};VS8;')
      WRITE(6,6) ESC
 6    FORMAT(1X,A1,'.)')
      RETURN
      END
      SUBROUTINE XAXIS11(ILINE,ITICK,IVAL,ICHAR,A,B,
     & IX1,IY1,IX2,IY2)
      CHARACTER*1 ESC/'{ESC}'/,CTLT/'{CTL-T}'/
      DIMENSION A(1),B(1)
      WRITE(6,1) ESC
 1    FORMAT(1X,A1,'.(')
      IF(ILINE.EQ.0) GOTO 10
      WRITE(6,2) IX1,IY1,IX2,IY1
 2    FORMAT(1X,'SP1;PA',(I5,',',I5),';PD',(I5,',',I5),';PU;')
C
 10   IF(ITICK.LE.1) GOTO 20
      DIST=(IX2-IX1)/(ITICK-1)
      WRITE(6,12) IX1,IY1
 12   FORMAT(1X,'SP1;PA',(I5,',',I5),';XT;')
      N=ITICK-1
      DO 13 I=1,N
 13   WRITE(6,14) DIST
 14   FORMAT(1X,'PR',F6.1,',0;XT;')
C
 20   IF(IVAL.LE.1) GOTO 30
      WRITE(6,22) IX1,IY1
 22   FORMAT(1X,'SP2;SI;PA',(I5,',',I5),';')
      DIST=-1.75*19*1.5*4
      WRITE(6,24) DIST
 24   FORMAT(1X,'PR',F6.0,',0;CP;')
      DIST=((IX2-IX1)/(IVAL-1))-4*19*1.5*4
      DO 27 I=1,IVAL
 27   WRITE(6,28) B(I),CTLT,DIST
 28   FORMAT(1X,'LB',A4,A1,';PR',F6.1,',0;')
C
 30    IF(ICHAR.EQ.0) GOTO 40
      N=ICHAR/4+1
      LENGTH=ICHAR*4*19*1.5
      AVAIL=(IX2-IX1)
      SIDE=(AVAIL-LENGTH)/2.
      WRITE(6,33) IX1,IY1,SIDE
 33   FORMAT(1X,'SP2;PA',(I5,',',I5),';SI;CP;CP;CP;PR',F6.0,',0;')
      WRITE(6,34) (A(I),I=1,16),CTLT
 34   FORMAT(1X,'LB',16A4,A1,';')
C
 40   WRITE(6,41)
 41   FORMAT(1X,'SP0;')
      WRITE(6,42) ESC
 42   FORMAT(1X,A1,'.)')
      RETURN
      END
      SUBROUTINE YAXIS85(ILINE,ITICK,IVAL,ICHAR,A,B,
     & IX1,IY1,IX2,IY2)
      CHARACTER*1 ESC/'{ESC}'/,CTLT/'{CTL-T}'/
```

Simulation Program - C-4

```
      DIMENSION A(1),B(1)
      WRITE(6,1) ESC
 1    FORMAT(1X,A1,'.(')
      IF(ILINE.EQ.0) GOTO 10
      WRITE(6,2) IX1,IY1,IX1,IY2
 2    FORMAT(1X,'SP1;PA',(I5,',',I5),';PD',(I5,',',I5),';PU;')
C
 10   IF(ITICK.LE.1) GOTO 20
      DIST=(IY2-IY1)/(ITICK-1)
      WRITE(6,12) IX1,IY1
 12   FORMAT(1X,'SP1;PA',(I5,',',I5),';YT;')
      N=ITICK-1
      DO 13 I=1,N
 13   WRITE(6,14) DIST
 14   FORMAT(1X,'PRO,'F6.1,';YT;')
C
 20   IF(IVAL.LE.1) GOTO 30
      WRITE(6,22) IX1,IY1
 22   FORMAT(1X,'SP2;SI;PA',(I5,',',I5),';')
      DIST=-(4*19*1.5*4+0.5*19*4)
      DIST2=-(.5*27*4)
      WRITE(6,24) DIST,DIST2
 24   FORMAT(1X,'PR',F6.0,',',F6.0,';')
      DIST=((IY2-IY1)/(IVAL-1))+27*2*4
      DO 27 I=1,IVAL
 27   WRITE(6,28) B(I),CTLT,DIST
 28   FORMAT(1X,'LB',A4,A1,';CP;PRO,'F6.1,';')
C
 30    IF(ICHAR.EQ.0) GOTO 40
      N=ICHAR/4+1
      LENGTH=ICHAR*4*19*1.5
      AVAIL=(IY2-IY1)
      SIDE=(AVAIL-LENGTH)/2.
      SHIFT=-6*19*1.5*4
      WRITE(6,33) IX1,IY1,SHIFT,SIDE
 33   FORMAT(1X,'SP2;PA',I5,',',I5,';SI;DI0,1;PR',F6.0,',0;PRO,',
     & F6.0,';')
      WRITE(6,34) (A(I),I=1,16),CTLT
 34   FORMAT(1X,'LB',16A4,A1,';')
C
 40   WRITE(6,41)
 41   FORMAT(1X,'SP0;DI;')
      WRITE(6,42) ESC
 42   FORMAT(1X,A1,'.)')
      RETURN
      END
      SUBROUTINE POINT(X,Y,IX1,IY1,IX2,IY2,ILINE,A1,N,SIZE)
      CHARACTER*1 ESC/'{ESC}'/,CTLT/'{CTL-T}'/
      DIMENSION X(1),Y(1)
      DATA PU,PD/'PU','PD'/
C        POINT ASSUMES PLOT IS TO BE IN USER-UNITS IN SPACE
C     ALREADY DEFINED BY P1 AND P2...
      WRITE(6,1) ESC
 1    FORMAT(1X,A1,'.(')
```

```
      WRITE(6,2) IX1,IX2,IY1,IY2,SIZE,SIZE,A1
2     FORMAT(1X,'SC',I5,',',I5,',',I5,',',I5,';SI',F4.2,',',F4.2,
     & ';SM',A1,';SP2;')
      ACT=PU
      DO 5 I=1,N
      WRITE(6,3) ACT,X(I),Y(I)
3     FORMAT(A2,';PA',F6.2,',',F6.2,';')
      IF(ILINE.NE.0) ACT=PD
5     CONTINUE
      WRITE(6,10)
10    FORMAT(1X,'PU;SP0;SI;SM;')
      WRITE(6,11) ESC
11    FORMAT(1X,A1,'.)')
      RETURN
      END
      SUBROUTINE TYPER
      CHARACTER*1 ESC/'{ESC}'/,CTLT/'{CTL-T}'/
      DIMENSION A(16)
      DATA ASTOP/'STOP'/
C     THIS VERSION OF TYPER TAKES CTL-T AS END-OF-STRING
      WRITE(6,2091)
2091  FORMAT('0YOU ARE NOW IN TYPER...')
      WRITE(6,2092) ESC
2092  FORMAT(1X,A1,'.(')
      WRITE(6,2094)
2094  FORMAT(1X,'SP2;')
21    WRITE(6,22) ESC
22    FORMAT(1X,A1,'.)')
      READ(5,23) A
23    FORMAT(16A4)
      IF(A(1).EQ.ASTOP) GOTO 35
      WRITE(6,146) ESC
146   FORMAT(1X,A1,'.(')
      WRITE(6,24) A
24    FORMAT(1X,'LB',16A4,'{CTL-T};CP;')
      GOTO 21
35    CONTINUE
      WRITE(6,3) ESC,ESC
3     FORMAT(1X,A1,'.(SP0;PA0,0;',A1,'.)')
      WRITE(6,4)
4     FORMAT(1X,' PROGRAM COMPLETED')
      RETURN
      END
      SUBROUTINE LINETP(I)
      IF(I.EQ.0) WRITE(6,1)
1     FORMAT(1X,'LT;')
      IF(I.GT.0) WRITE(6,2) I
2     FORMAT(1X,'LT',I1,';')
      RETURN
      END
      SUBROUTINE PLOTOF
      CHARACTER*1 ESC/'{ESC}'/,CTLT/'{CTL-T}'/
      WRITE(6,1) ESC
1     FORMAT(1X,'SP0;')
```

```
      WRITE(6,2)
 2    FORMAT(1X,A1,'.)')
      RETURN
      END
      SUBROUTINE PLOTON
      CHARACTER*1 ESC/'{ESC}'/,CTL'/'{CTL-T}'/
      WRITE(6,1) ESC
 1    FORMAT(1X,A1,'.(')
      RETURN
      END
```